PLC based fractional-order PID temperature control in pipeline: design procedure and experimental evaluation

Jakub Możaryn · Jakub Petryszyn · Stepan Ozana

Received: 5 March 2020 / Accepted: 18 July 2020 / Published online: 3 August 2020
© The Author(s) 2020

Abstract Fractional-order control system design can be used for systems with non-local dynamics involving long-term memory effects. However, implementation of a fractional-order controller in industrial systems is complicated, because of the excessive demand for computational power. The following paper presents the step-by-step design procedure, parameter tuning, and experimental evaluation of the fractional order proportional-integral-derivative (FOPID) controller. The control algorithm is based on the Continued Fraction Expansion approximation of the fractional-order operators. It is implemented on a standard industrial Programmable Logic Controller. The FOPID control system is verified and evaluated in terms of efficiency and robustness using a new laboratory benchmark of a temperature control in the pipeline. The proposed solution shows increased efficiency in terms of robustness compared to the standard PID closed-loop control.

Keywords Fractional order PID · Temperature control · Continued fraction expansion · Programmable logic controller

1 Introduction

The Proportional-Integral-Derivative (PID) algorithm is one of the most used algorithms in the industry to control processes. It has a simple implementation and there is a vast number of easy-to-use and intuitive tuning methods for such a controller depending on the controlled process with required control quality [20, 38, 48].

Many physical phenomena exhibit complex microscopic behaviours. Most of the processes associated with complex systems have non-local dynamics involving long-term memory effects. To model such processes with higher accuracy, fractional calculus, i.e. the integration and differentiation of the arbitrary order, can be used [28, 32, 39]. Fractional derivative operators are non-local. This means that a functional derivative of function $f(x, t)$ depends on all the values of this function in the interval $[t_0, t]$ (entire history of $f(x, t)$). Some examples of using fractional calculus for modelling physical processes are e.g. acoustics, diffusive transport, fluid flow, geodynamics, optics, chemical engineering and electromagnetic waves [16].

J. Możaryn (✉) · J. Petryszyn
Faculty of Mechatronic, Institute of Automatic Control
and Robotics, Warsaw University of Technology, Ul. Św
A. Boboli 8, 02-525 Warsaw, Poland
e-mail: jakub.mozaryn@pw.edu.pl

S. Ozana
Department of Cybernetics and Biomedical Engineering,
Faculty of Electrical Engineering and Computer Science,
VSB-Technical University of Ostrava, 17. Listopadu
2172/15, 708 00 Ostrava, Czech Republic
e-mail: stepan.ozana@vsb.cz

The standard PID control algorithm and dynamic models used in control engineering are based on integer order calculus (a special case of fractional-order calculus) [8]. It can be shown that integer derivative operators don't hold the non-locality property [11]. To effectively control the dynamic systems described by fractional-order equations, fractional-order control algorithms should be designed. The major problem with control engineering applications of fractional calculus was the complicated method of solving the differential equations. It was overcome with finding the proper geometric and physical interpretation of the fractional operators [56] and their efficient numerical approximations, like Power Series Expansion (PSE) [54] or Continued Fraction Expansion (CFE) [7], among others. One family of such controllers is the set of Fractional Order PID-type (FOPID) controllers, such as PID^k [31], CRONE (fr. *Contrôle Robuste d'Ordre Non Entrier*, Non-Integer Order Robust Control) [47], and $PI^\lambda D^\mu$ [55]. Such controllers have increased the number of tuned parameters from 3 to 5 (compared with standard PID) and allow better matching of the controller with the process. There are also different tuning methods proposed for the FOPID controller which fulfill the control system design requirements e.g. optimization with integral criteria [9, 32], constrained min-max optimization [5], swarm optimization [21], auto-tuning methods [33], and robust tuning methods [59]. In [9, 32, 65] authors show that the FOPID controller outperforms the standard PID controller in applications where processes exhibit non-local dynamics.

In the literature, many practical applications of the FOPID controller and its variants are proposed. In [10] authors present FOPI controller for wind turbine generators. Applications of FOPID control in electro-hydraulic systems are considered in [18, 60, 66]. In [3] the twin-rotor helicopter control system based on FOPID controller is described. The optimal tilt FOPID control in rail vehicles is presented in [15]. In [27, 35] the implementation of FOPI controller in industrial electric drives (DC-motors and PMSM drives) is described. Application of FOPID control in the pump-turbine governing system (PTGS) is proposed in [26]. FOPID control of the Automatic Voltage Regulator (AVR) is given in [50, 64]. In [61] FOPID control system for pumped storage hydro unit is described.

FOPID fuzzy control of the combined cycle power plant is proposed in [14]. In [25] the FOPID controller is implemented in the pneumatic servo-system. In [58] the FOPID controller is applied in Load Frequency Control (LFC) in power systems. In [29] a decoupled visual model and a FOPID controller are designed aiming at the problem of the view distortion in the fillet weld welding process.

Most FOPID algorithms are usually tested and simulated in a MATLAB-Simulink environment [10, 14, 15, 26, 50, 58, 61, 62, 64]. Controllers that implement the fractional-order control algorithms usually are equipped with fast and efficient DSP processors (e.g. dSpace¹), because extensive computations are required [18, 27, 29, 32, 35, 66]. It makes such a solution hard to apply in the industry. The tuning of controller parameters is also a problem that needs to be solved to ensure the proper functionality of the automation system. Nowadays, most production lines in small and medium companies are controlled by Programmable Logic Controllers (PLC). Therefore, PLC implementation is favorable in industrial applications. In [23] authors present the PLC implementation of a CRONE 3rd generation controller for a hydraulic system based on SIEMENS S7-300 PLC. The PLC implementation of the FOPID algorithm with CFE approximation for temperature control using a BR 2005 controller is described in [51]. The FOPID algorithm with PSE approximation for the water level in a hydraulic system is given in [36]. Implementations of fractional operators based on CFE approximation using SIEMENS S7-1200 PLC are presented in [42], and Hardware-in-the-Loop (HIL) experiments are described in [43]. Finally, preliminary results of laboratory benchmark for robust PLC-based FOPID temperature control in the pipeline is presented in [53].

Many solutions presented in the literature on the effective PLC FOPID implementation, are tested in the Hardware-in-the-Loop setup [36, 42–45]. Only few describe the laboratory experiments with real processes [25, 36, 51]. This article synthesizes the mentioned works and describes the step-by-step design procedure, parameter tuning, and implementation of the $PI^\lambda D^\mu$ algorithm with CFE approximation on a laboratory test stand. The control system is equipped with a Siemens S7-1200 PLC for control and

¹ <https://www.dspace.com/>.

data acquisition. The controlled value is the temperature in the pipeline with induced air-flow. Moreover, on the test stand, it is possible to impose additional disturbances, to check the robustness of the proposed solution. In the paper the comprehensive tests, covering both the set-point tracking and the disturbance rejection, are considered, which is rarely mentioned in the articles about industrial PLC implementations of FOPID controllers. Therefore, robustness of the PI^2D^μ algorithm can be thoroughly tested. The proposed solution shows increased efficiency in terms of robustness compared to the standard PID control.

The article is organized as follows: In Sect. 2, basic fractional-order calculus definitions are given. Then, in Sect. 3 the PI^2D^μ controller algorithm and its parameters' tuning are described. In Sect. 4 numerical approximation of fractional operators is explained. Next, in Sect. 5, the laboratory test stand is described. In Sect. 6 the PLC implementation of the PI^2D^μ controller is presented. In Sect. 7 experimental results for the PI^2D^μ controller, along with a comparison with the standard PID controller are shown and discussed. Finally, concluding remarks are given in Sect. 8.

2 Fractional order calculus definitions

2.1 Differ-integral operator

The fractional calculus generalizes the well-known integer order actions by using real numbers for differentiation and integration actions. The differ-integral operator is the combination of non-integral differentiation and integration operators, defined as [11]

$${}^{\text{Def}}_a \mathfrak{D}_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha}, & \alpha > 0 \\ 1, & \alpha = 0 \\ \int_a^t (d\tau)^{-\alpha}, & \alpha < 0 \end{cases} \quad (1)$$

where α —order of the operation ($\alpha \in \mathbb{R}$), a and t are the lower and upper limits of integration.

The differ-integral operator (1) acts as the integrator for $\alpha < 0$ and as the derivative for $\alpha > 0$.

The Riemann–Liouville definition (RL definition) of a differ-integration of function $f(t)$ is given as [11]

$$\begin{aligned} {}^{\text{RL}}_a \mathfrak{D}_t^\alpha f(t) &= \frac{d^\alpha}{dt^\alpha} f(t) \\ &= \frac{1}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_0^t f(\tau)(t-\tau)^{m-\alpha-1} d\tau, \end{aligned} \quad (2)$$

where $m-1 \leq \alpha \leq m$, $m \in \mathbb{N}$ —Riemann–Liouville fractional derivative-integral, $\Gamma(\cdot)$ —Euler Gamma function [19], defined as

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt, \quad \Re(x) > 0. \quad (3)$$

The most common differ-integral operator definitions are the Riemann–Liouville (RL definition) [11], the Caputo (C definition) [6], and the Grünwald–Letnikov definition (GL definition) [28, 32].

Because of the need of numerical implementation in PLC controllers, usually the Grünwald–Letnikov definition of a differ-integration of the function $f(t)$ is given as

$$\begin{aligned} {}^{\text{GL}}_a \mathfrak{D}_t^\alpha f(t) &= \frac{d^\alpha f(t)}{d(t-a)^\alpha} \\ &= \lim_{h \rightarrow 0} h^{-\alpha} \sum_{j=0}^{\lfloor \frac{t-a}{h} \rfloor} (-1)^j \binom{\alpha}{j} f\left(t - j \left[\frac{t-a}{h} \right]\right), \end{aligned} \quad (4)$$

where

$$\binom{\alpha}{j} = \begin{cases} 1, & j = 0 \\ \frac{\alpha(\alpha-1)\dots(\alpha-j+1)}{j!}, & j > 0 \end{cases} \quad (5)$$

is a generalization of the Newton symbol into real numbers.

For numerical simulations, the GL definition (4) can be rewritten as

$$\begin{aligned} {}^{\text{GL}}_a \mathfrak{D}_t^\alpha f(t) &= \frac{d^\alpha f(t)}{d(t-a)^\alpha} \\ &= \lim_{h \rightarrow 0} h^{-\alpha} \sum_{j=0}^{\lfloor \frac{t-a}{h} \rfloor} w_j^{(\alpha)} f\left(t - j \left[\frac{t-a}{h} \right]\right), \end{aligned} \quad (6)$$

where $w_j^{(\alpha)}$ —binominal coefficients, calculated recursively as follows

$$w_0^{(\alpha)} = 1, \quad w_j^{(\alpha)} = \left(1 - \frac{\alpha + 1}{j}\right) w_{j-1}^{(\alpha)}, \quad j = 1, 2, \dots \quad (7)$$

It should be emphasized, that RL (2) and GL (4) formulations are equivalent [30].

The Laplace transform of (2) can be calculated as [54]

$$\mathcal{L}[\text{RL}_0 \mathfrak{D}_t^\alpha f(t)] = s^\alpha F(s) - \sum_{k=0}^{m-1} s^k [\text{RL}_0 \mathfrak{D}_t^{\alpha-k-1} f(t)]_{t=0}, \quad (8)$$

where $F(s)$ —Laplace transform of the function $f(t)$, s —complex variable, $m > 0$, $m - 1 \leq \alpha < m$

2.2 Oustaloup approximation of differ-integral operator

Direct numerical implementation of the differ-integral operator (4) is difficult to apply, and it is necessary to approximate it. The commonly used approximation method in practical implementations is the Oustaloup filter in the form of a product of a series of stable first-order linear systems [4, 46]

$$s^\alpha = C \prod_{i=1}^n \frac{s + \omega'_i}{s + \omega_i}. \quad (9)$$

The Oustaloup filter coefficients are calculated as follows

$$C = \omega_g^\alpha, \quad (10)$$

$$\omega'_i = \omega_d \omega_w^{(2i-1-\alpha)/n}, \quad (11)$$

$$\omega_i = \omega_d \omega_w^{(2i-1+\alpha)/n}, \quad (12)$$

$$\omega_w = \sqrt{\frac{\omega_g}{\omega_d}}, \quad (13)$$

where ω_d , ω_g —lower and higher bound frequencies, n —the order of the approximation.

3 $\text{PI}^\lambda \text{D}^\mu$ controller algorithm and tuning

The proposed $\text{PI}^\lambda \text{D}^\mu$ controller transfer function is in the following form [55]

$$G_{\text{FOPID}}(s) = K_{P_{\text{FOPID}}} + T_{I_{\text{FOPID}}} s^\lambda + T_{D_{\text{FOPID}}} s^\mu, \quad (14)$$

where $K_{P_{\text{FOPID}}}$ —the proportional gain, $T_{I_{\text{FOPID}}}$ —the integral time constant, $T_{D_{\text{FOPID}}}$ —the derivative time constant, s^λ , s^μ —differ-integral operators ($\lambda < 0$, $\mu > 0$).

The block diagram of the controller structure is given in Fig. 1, where the input of the controller is the difference between the measured process value and set-point value given as

$$e(t) = y_{\text{SP}}(t) - y_m(t), \quad (15)$$

where $y_m(t)$ —the measured plant output, $y_{\text{SP}}(t)$ —set-point.

$\text{PI}^\lambda \text{D}^\mu$ controller can be tuned according to the optimization method described in [49, 52, 65], assuming the controller transfer function in the form (14) and the process transfer function in the form

$$G_o(s) = \frac{1}{a_1 s^\alpha + a_2 s^\beta + a_3}. \quad (16)$$

The chosen tuning method is based on the Oustaloup approximation (9). It requires information about the desired gain margin A_m and phase margin θ_m of the closed-loop control system to solve the following set of equations

$$K_{P_{\text{FOPID}}} + \frac{T_{I_{\text{FOPID}}}}{\omega_p^\lambda} \cos \frac{\pi \lambda}{2} + T_{D_{\text{FOPID}}} \omega_p^\mu \cos \frac{\pi \mu}{2} = -\frac{a_1}{A_m} \omega_p^\alpha \cos \frac{\pi \alpha}{2} - \frac{a_2}{A_m} \omega_p^\beta \cos \frac{\pi \beta}{2} - \frac{a_3}{A_m}, \quad (17)$$

$$-\frac{T_{I_{\text{FOPID}}}}{\omega_p^\lambda} \sin \frac{\pi \lambda}{2} + T_{D_{\text{FOPID}}} \omega_p^\mu \sin \frac{\pi \mu}{2} = -\frac{a_1}{A_m} \omega_p^\alpha \sin \frac{\pi \alpha}{2} - \frac{a_2}{A_m} \omega_p^\beta \sin \frac{\pi \beta}{2}, \quad (18)$$

Fig. 1 Block diagram of the $\text{PI}^\lambda \text{D}^\mu$ controller

$$\begin{aligned}
 K_{P_{\text{FOPID}}} + \frac{T_{I_{\text{FOPID}}}}{\omega_g^\lambda} \cos \frac{\pi\lambda}{2} + T_{D_{\text{FOPID}}} \omega_g^\mu \cos \frac{\pi\mu}{2} \\
 = -a_1 \omega_g^\alpha \cos \left(\frac{\pi\alpha}{2} + \theta_m \right) - a_2 \omega_g^\beta \cos \left(\frac{\pi\beta}{2} + \theta_m \right) \\
 - a_3 \cos \theta_m,
 \end{aligned} \quad (19)$$

$$\begin{aligned}
 -T_{I_{\text{FOPID}}} \frac{\sin \frac{\pi\lambda}{2}}{\omega_g^\lambda} + T_{D_{\text{FOPID}}} \sin \frac{\pi\mu}{2} \omega_g^\mu \\
 = -a_1 \omega_g^\alpha \sin \left(\frac{\pi\alpha}{2} + \theta_m \right) - a_2 \omega_g^\beta \sin \left(\frac{\pi\beta}{2} + \theta_m \right) \\
 - a_3 \sin \theta_m.
 \end{aligned} \quad (20)$$

where ω_g , ω_p —the gain and phase crossover frequencies.

To calculate ω_p , ω_g the following condition has to be fulfilled

$$\begin{aligned}
 \left(\omega_g^{\lambda+\mu} - \omega_p^{\lambda+\mu} \right) \left\{ a_1 \left[\omega_g^\alpha \cos \left(\frac{\pi\alpha}{2} + \theta_m \right) - \omega_p^\alpha \cos \frac{\pi\alpha}{2} \right] \right. \\
 + a_2 \left[\omega_g^\beta \cos \left(\frac{\pi\beta}{2} + \theta_m \right) - \omega_p^\beta \cos \frac{\pi\beta}{2} \right] \\
 + a_3 \cos \left(\theta_m - \frac{1}{A_m} \right) \left. \right\} \\
 + \left(\text{ctg} \frac{\pi\lambda}{2} + \text{ctg} \frac{\pi\mu}{2} \right) \left(\frac{\omega_p^\lambda \omega_g^\mu F_p}{A_m} + \omega_g^\beta \omega_p^\mu F_g \right) \\
 - \left(\frac{\omega_p^{\lambda+\mu} F_p}{A_m} + \omega_g^{\lambda+\mu} F_g \right) \text{ctg} \frac{\pi\mu}{2} \\
 - \left(\omega_p^{\lambda+\mu} F_g + \frac{\omega_p^{\lambda+\mu} F_p}{A_m} \right) \text{ctg} \frac{\pi\lambda}{2} = 0,
 \end{aligned} \quad (21)$$

$$F_p = a_1 \omega_p^\alpha \sin \frac{\pi\alpha}{2} + a_2 \omega_p^\beta \sin \frac{\pi\beta}{2}, \quad (22)$$

$$\begin{aligned}
 F_g = a_1 \omega_g^\alpha \sin \left(\frac{\pi\alpha}{2} + \theta_m \right) + a_2 \omega_g^\beta \left(\sin \frac{\pi\beta}{2} + \theta_m \right) \\
 + a_3 \sin \theta_m.
 \end{aligned} \quad (23)$$

During the controller parameters calculations, to fulfill the conditions (21)–(23), the numerical optimization methods are usually used. Then, if the parameters ω_p , ω_g , λ , μ are given, the controller gains $K_{P_{\text{FOPID}}}$, $T_{I_{\text{FOPID}}}$, $T_{D_{\text{FOPID}}}$ can be calculated from (17)–(20) as follows

$$\begin{aligned}
 K_{P_{\text{FOPID}}} = a_1 \omega_p^{\lambda+\alpha} \frac{\sin \frac{\pi(\alpha-\mu)}{2}}{A_m \sin(\pi\mu/2)} \\
 + a_1 \omega_g^{\lambda+\alpha} \frac{\sin \left[\frac{\pi(\mu-\alpha)}{2} - \phi_m \right]}{\sin(\pi\mu/2)} \\
 + a_2 \omega_p^{\lambda+\beta} \frac{\sin \left[\frac{\pi(\beta-\mu)}{2} \right]}{A_m \sin(\pi\mu/2)} \\
 + a_2 \omega_g^{\lambda+\beta} \frac{\sin \left[\frac{\pi(\mu-\beta)}{2} - \phi_m \right]}{\sin(\pi\mu/2)} - \frac{a_3 \omega_p^\lambda}{A_m} \\
 + a_3 \omega_g^\lambda \frac{\sin \left(\frac{\pi\mu}{2} - \phi_m \right)}{\sin(\pi\mu/2)} - \frac{a_3 \omega_p^\lambda}{A_m},
 \end{aligned} \quad (24)$$

$$T_{I_{\text{FOPID}}} = \frac{\omega_g^\lambda \omega_p^\lambda (A_m \omega_g^\mu F_p - \omega_p^\mu F_g)}{\sin \frac{\pi\lambda}{2} (\omega_g^{\lambda+\mu} - \omega_p^{\lambda+\mu})}, \quad (25)$$

$$T_{D_{\text{FOPID}}} = \frac{A_m \omega_p^\lambda F_p - \omega_g^\lambda F_g}{\sin \frac{\pi\mu}{2} (\omega_g^{\lambda+\mu} - \omega_p^{\lambda+\mu})}. \quad (26)$$

4 Discrete approximation of differ-integral

To implement the differ-integral operator in the digital controller (e.g. PLC), different discrete approximation methods are used. The first common approximation method is the Power Series Expansion (PSE) [42] that implements directly GL differ-integral. PSE leads to a solution in the form of polynomials, that have a digital Finite Impulse Response (FIR) filter structure. Such an approximation of the differ-integral operator depends only on the zeroes of the filter, so only system outputs are considered in the solution. Therefore, reasonable PSE approximation requires output signal values from many previous steps, which increases the requirements for the controller's hardware and memory.

Another approximation method is the Continued Fraction Expansion (CFE) [40, 42]. CFE approximation leads to the solution in the form of rational functions. Therefore, it has an Infinite Impulse Response (IIR) filter structure. It depends on both zeros and poles of the system, so it requires previous outputs and inputs of the system. With CFE, there is a lower requirement for output and input signal values from previous steps required than with PSE. It allows for fast convergence and low order of the filter.

For the discrete approximation of the fractional operator, the generating function is used in the form

$$\omega(z^{-1}) = 2 \frac{1 - z^{-1}}{1 + z^{-1}}, \quad (27)$$

where z —complex variable.

Then, the CFE approximation of the fractional operator can be written as follows

$$s^{\pm\alpha} \approx (\omega(z^{-1}))^{\pm\alpha} = \left(2 \frac{1 - z^{-1}}{1 + z^{-1}} \right)^{\pm\alpha}. \quad (28)$$

The discrete transfer function, approximating the fractional operator is given as

$$\begin{aligned} D^{\pm\alpha} &= \frac{Y(z)}{F(z)} = (\omega(z^{-1}))^{\pm\alpha} \\ &= \left(\frac{1+a}{T_p} \right)^{\pm\alpha} \text{CFE} \left| \left(\frac{1 - z^{-1}}{1 + az^{-1}} \right)^{\pm\alpha} \right|_{p,q} \\ &= \left(\frac{1+a}{T_p} \right)^{\pm\alpha} \frac{P_p(z^{-1})}{Q_q(z^{-1})} = \left(\frac{1+a}{T_p} \right)^{\pm\alpha} \frac{\sum_{i=0}^p P_i z^{-i}}{\sum_{i=0}^q Q_i z^{-i}}, \end{aligned} \quad (29)$$

where $a \in [0, 1]$ —argument that depends on the approximation method ($a = 0$ for Euler approximation and $a = 1$ for Tustin approximation), T_p —sampling period, $Y(z)$ —discrete transform of the output sequence $y(nT_p)$, $F(z)$ —discrete transform of the input sequence $f(nT_p)$, p, q —orders of the approximation (usual assumption is $p = q = n$, so then n is the approximation order).

The time response of the element (29) is expressed as

$$y_{\text{CFE}}(k) = \frac{1}{v_0} \left[- \sum_{i=1}^n v_i y(k-i) + \sum_{j=0}^n w_j u(k-j) \right], \quad (30)$$

where $y(k)$, $u(k)$ —the output and input signals in provided sequence, v_i and w_j —coefficients of CFE approximation (for exact values see [40]), k —discrete time, $t = kT_p$.

Since CFE provides better memory usage, it can be used for approximation of fractional operators in real systems with limited capacity.

5 Description of laboratory test stand

The process part of the laboratory test stand is presented in Fig. 2, whereas the schematic diagram is given in Fig. 3.

The laboratory test stand comprises a pipeline with a heater (G) in which airflow is induced by a fan (S). During experiments, the control system changes the amount of heat released by the heater (G) and maintains constant air-flow. The power control signal of the heater (Y_G) and speed of the fan (Y_W) are standard current signals 4–20 mA generated by a PLC controller used in a closed-loop control system.

To measure the temperature a T/I measuring transducer with a Pt100 resistance thermometer and additional linearization was used. The measurement range of the transducer was 25°C - 75°C (output standard current signal 4–20 mA). To control the system, a PLC S7-1212c Siemens controller was used with an analog input/output module and a 24V power supply. A SIMATIC HMI KPT 600 panel and desktop PC with TIA-Portal software were used for visualization and controller implementation.

It was possible to artificially induce the following disturbances:

- a step-change in the air inlet orifice - change of the cross-section from 389 to 1661 mm² (DI1);
- a step-change in the heater power (G) by adding or disconnecting additional resistance which changes the heater resistance from 100 Ω to 75 Ω (DI2).

Fig. 2 The photo of the laboratory test stand: **a** PC with the TIA-Portal software, **b** the control interface, **c** the pipeline with induced airflow and the heater, **d** the industrial box with the PLC controller and the HMI panel for the process monitoring

Fig. 3 The schematic diagram of the PLC-based temperature control system: **a** PLC controller, **b** HMI panel for the process monitoring, **c** Industrial Ethernet communication link, **d** PC with the TIA-Portal software, **e** the pipeline with the heater and induced air-flow

All the values of the control signal (CV), the process signal (PV), and the set-point (SP) were scaled to a range 0–100 %, to simplify further calculations and discussion of the experimental evaluation results.

6 PLC Implementation of PI^1D^μ controller

The design and PLC implementation procedure of the PI^1D^μ (FOPID) controller is presented in Fig. 4. It consists of 7 steps described below.

6.1 Process identification

The mathematical model of the process was approximated by the transfer function of the First Order Plus Dead Time (FOPDT) system.

$$G_{ob}(s) = \frac{PV(s)}{CV(s)} = \frac{K_{ob}}{T_z s + 1} e^{-T_0 s}, \quad (31)$$

where $PV(s) = y(s)$ —the process value, $CV(s) = u(s)$ —the control value, T_z —the process time constant, T_0 —the process dead time, K_{ob} —the process gain.

The plant parameters were determined from the open-loop system step response using the secant method [12]. The FOPDT model parameters were calculated as follows

STEP 1

Define the structure of the FOPID controller

STEP 2

Identify the transfer function of the controlled process

STEP 3

Define the performance parameters of the control system

STEP 4

Tune the parameters of the FOPID controller

STEP 5

Choose the approximation method of fractional operator

STEP 6

Implement the FOPID controller in Simulink environment

STEP 7

Transfer the FOPID controller to PLC

Fig. 4 Design and implementation procedure

$$K_{ob} \approx \frac{\Delta PV}{\Delta CV}, \quad T_0 \approx \frac{T_1 - T_2 \ln(2)}{1 - \ln(2)}, \quad T_z \approx T_2 - T_0, \quad (32)$$

where ΔCV —step change of CV, ΔPV —amplitude of the step response, T_1 , T_2 —time constants for the step response to reach 50% and 63,2% of its final value.

Then, the Second Order model of the process was calculated. It allows for getting the transfer function in the form (16), based on the FOPDT transfer function (31). The Second Order model parameters can be approximated as follows [22]

$$a_1 = \frac{T_a T_b}{K_{ob}}, \quad a_2 = \frac{T_a + T_b}{K_{ob}}, \quad a_3 = \frac{1}{K_{ob}}, \quad (33)$$

where

$$T_a \approx T_z - T_0, \quad T_b \approx 2T_0. \quad (34)$$

The goodness of fit measure between the estimated output signal $y_m(k)$ and the reference output signal $y_r(k)$ was calculated as Normalized Root Mean Square Error [17]

$$\text{NRMSE} = 1 - \frac{\|y_{\text{est}}(k) - y_r(k)\|}{\|y_r(k) - \bar{y}_r\|}, \quad (35)$$

where $y_{\text{est}}(k)$ —estimated output signal, $y_r(k)$ —reference signal, $\bar{y}_r$ —mean value of the reference signal, k —sample. Value of NRSME varies between $-\infty$ (bad fit) to 1 (perfect fit).

Both identified models, with corresponding NRMSE values, are gathered in Table 1. In Fig. 5 step responses are presented and in Fig. 6 estimation errors for identified models are presented. For both models quality index is around 0.98, which indicates a high modeling accuracy.

6.2 Tuning of the controller

While tuning the $\text{PI}^\lambda \text{D}^\mu$ controller, the following requirements were placed:

- no steady-state error;
- gain margin: $A_m = 10$ dB;
- phase margin: $\theta_m = 60^\circ$;
- identified parameters of the Second Order model (see Table 1): $a_1 = 203.3639$, $a_2 = 59.2654$, $a_3 = 1.3014$, $\alpha = 2$, $\beta = 1$.

The Interior-Point optimization method [37] with the Integral Squared Time Error (ITSE) criterion was used to fulfill the conditions (21)–(23). The calculated parameters of the $\text{PI}^\lambda \text{D}^\mu$ controller, according to (24)–(26), are gathered in Table 2.

Table 1 Accuracy of identified models ($G_{\text{obl}}(s)$ - First Order Plus Dead-Time (FOPDT), $G_{\text{obl2}}(s)$ - Second Order (SO))

Transfer function	NRMSE
$G_{\text{obl}}(s) = \frac{0.765}{43.6691s + 1} e^{-1.8691s}$	0.9839
$G_{\text{obl2}}(s) = \frac{1}{203.3639s^2 + 59.2654s + 1.3014}$	0.9802

Fig. 5 Step responses comparison for the identified models

Fig. 6 Estimation errors for the identified models

Table 2 $\text{PI}^\lambda \text{D}^\mu$ controller settings

$K_{\text{P}_{\text{FOPDT}}}$	$T_{\text{I}_{\text{FOPDT}}}$	$T_{\text{D}_{\text{FOPDT}}}$	λ	μ
2.1420	0.8560	29.626	-0.9851	0.7998

6.3 Controller implementation

The implementation process of the $\text{PI}^\lambda \text{D}^\mu$ control algorithm (14) on PLC can be divided into two phases.

6.3.1 Phase 1: simulink implementation

In the first phase, the $\text{PI}^\lambda \text{D}^\mu$ control algorithm is implemented in the MATLAB/Simulink environment. The set of MATLAB functions and functional Simulink blocks should be created.

In Fig. 7 the overall Simulink diagram of the implemented $\text{PI}^\lambda \text{D}^\mu$ controller is given.

In the block CFE (see Fig. 8) the Continued Fraction Expansion approximations of the fractional operators s^λ and s^μ are implemented. In blocks I_fopid and D_fopid (see Figs. 9, 10) the

Fig. 7 Simulink diagram of the $PI^\lambda D^\mu$ controller

Fig. 8 Simulink diagram of the Continued Fraction Expansion approximations of the fractional operators s^λ and s^μ (CFE block)

Fig. 9 Simulink diagram of the fractional integral action

fractional integral and derivative actions are implemented. The block P_fopid (see Fig. 7) realizes the controller transfer function (14).

6.3.2 Phase 2: transfer to PLC

In the second phase, the code for the PLC is implemented in the Structured Control Language

Fig. 10 Simulink diagram of the fractional derivative action

(SCL). For this purpose, Simulink PLC Coder² can be efficiently used, which allows code creation for PLC directly from the MATLAB code.

The program in the TIA Portal environment (Siemens) can be divided into blocks. In this way, a modular structure of the application is obtained. There are 4 types of blocks:

- **Organizational Blocks (OB)** allow the code to run when a specific condition occurs. The condition may be, for example, a measured period, a specific date, error, or hardware configuration change.
- **Function Blocks (FB)** the instructions they contain are also executed cyclically. However, the instructions they contain usually need over one program loop cycle. Therefore, it is often necessary to know the function values from the previous cycles. For this purpose, data blocks (DB) are used.
- **Functions (FC)** used to perform simple and repetitive instructions such as assignment instructions or simple arithmetic calculations,
- **Data Blocks (DB)** these blocks make it possible to store data determined during controller operation, and are also available in every other block.

Implemented TIA program structure, with Function Blocks realizing FOPID algorithm and accompanying Data Blocks, is presented in Fig. 11.

The controller code has been divided into 5 function blocks. The $fbCFE$ determines the coefficients according to the CFE approximation described. It takes the sampling period, fractional integration factor λ , fractional derivative factor μ , and the factor determining the discretization method as the parameters. As a result of this function, 4 tables are created that are the weighting factors of the subsequent inputs

² <https://www.mathworks.com/products/sl-plc-coder.html>.

Fig. 11 TIA program structure of the implemented FOPID controller

and outputs of the fractional integrating and differentiating actions. Then, based on the received tables and the input state table, the integral values in block fbIntegral and derivative values in block fbDerivative are determined. Finally, the controller response is calculated in block fbFOPID. For the program to work, it was also necessary to implement an organizational block that starts at the time of starting the controller to determine the initial conditions and the block calling cyclic interrupts with a frequency equal to the assumed sampling frequency.

7 Experimental evaluation of controller

7.1 Quality indices

During tests of the controller, the quality of control has to be evaluated. It was analyzed in the time domain using the following indices [34]

- *Steady-state error* e_{st} - error in steady-state after introducing the set-point change or the disturbance.
- *Maximum error*

$$e_{\max} = \max_T |e(t)| \quad (36)$$

where $T \in [T_{sp}, T_R]$ —it is the time-span between the T_{sp} —first moment of reaching the set point value and settling time T_R . If $T_{sp} > T_R$, $e_{\max} = 0$.

- *Overshoot*

$$\gamma = \left| \frac{e_2}{e_1} \right| 100\% \quad (37)$$

where e_1 and e_2 are the first two consecutive biggest errors with opposite signs, assuming as the zero level (baseline) the steady-state value of the output signal $y(t)$ after the transient response.

- *Settling time* T_R —it is the time between the moment of change of the set point $y_{sp}(t)$, or introduction of disturbances $d(t)$ and the moment when error $e_{sp}(k)$ reaches a fixed value inside a boundary e.g. $\pm 0.1\%$.

7.2 Selection of the sampling period

Digital devices such as a PLC or a micro-controller have to work with a finite sampling period (T_p). Thus, the process output is available only at the sample points and the process variable is discrete. To guarantee the expected performance in terms of the continuous plant output it is crucial to choose an adequate value of T_p . When discretizing continuous controllers by using mathematical formulas describing the relation between Laplace and Z-transforms, a smaller sampling period means higher accuracy. Numerous research was done in this topic (considering constant and variable sampling rates) [24, 41, 63]. For the FOPDT system (31), the following practical rule is proposed [2]

$$T_p \approx \frac{T_z}{N_r} \quad (38)$$

where N_r —a number of sampling periods during T_z (rise time).

In PLC implementations, the value of N_r is usually chosen from the range 4–10 [2]. The equation (38) allows calculating the maximum sampling period. For the process considered in this work $T_z = 43.6691$ s (see Table 1), thus the sampling period should be lower than 4 s ($N_r = 10$). There is no theoretical lower limit for the sampling period for the considered controller. However, PLC with a too-short T_p cannot provide all required calculations in real-time, and better performance, though it can lead to unnecessary

costs of high-end instrumentation and capacity resources. On the other hand, a low sampling rate will reduce the control quality because the controller's response to interference is too slow.

We have chosen the sampling period $T_p = 200$ ms for PID and FOPID for simulations and PLC implementation. It should be mentioned, that sampling in the Siemens S7-1200 PLC is not fixed (as in MATLAB/Simulink), and there exist ± 0.005 s margin. This T_p value was chosen concerning the requirements for computing performance (with the I/O manipulation, HMI communication or data monitoring and acquisition) and control quality. There was observed, that increasing further the sampling rate decreased quality because some of the control cycles didn't produce proper control value.

7.3 Simulations

During simulations, the algorithm implemented in the PLC was used to control the process simulated in MATLAB/Simulink environment (Software-In-the-Loop, SIL). For this purpose, it was necessary to transfer data between the Siemens PLC software (TIA Portal) and the MATLAB/Simulink environment. An additional OPC UA server was used to enable data exchange between the simulated controller in the PLCsim (part of the TIA-Portal software) and Simulink. NetToPLCsim software³ extension allowed TCP / IP connections of PLC with the simulated process. Calculated quality indices for SIL tests were gathered in Table 3. This confirmed the effective transfer of code from the MATLAB/Simulink environment to the Siemens S7-1200 PLC controller.

7.4 Laboratory tests

During laboratory tests, the designed and tuned $PI^{\lambda}D^{\mu}$ controller was used in the laboratory test stand. The step response of the control system with the $PI^{\lambda}D^{\mu}$ controller, and the influence of disturbances were examined.

For the comparison, the integer-order PID controller was chosen, in the form

Table 3 Quality of the $PI^{\lambda}D^{\mu}$ temperature control system for the Software-In-the-Loop simulation

	$e_{\max}$ [%]	e_{st}	γ [%]	T_R [s]
Set-point change	1.00	0.00	8.50	87.00

$$G_{PID}(s) = K_{PID} \left(1 + \frac{1}{T_{PID}s} + \frac{T_{D_{PID}}s}{\frac{T_{D_{PID}}}{N_D}s + 1} \right), \quad (39)$$

where K_{PID} —proportional gain, T_{PID} —integral time constant, $T_{D_{PID}}$ —derivative time constant and $\frac{T_{D_{PID}}}{N_D}$ —time constant of first-order filter ($N_D \in [5, 20]$, $N_D \in R^+$ [1]).

The integer-order PID controller was tuned with a Ziegler-Nichols method [67], which is well known among control engineers. These tuning rules give significant disturbance rejection. However, the drawback is the possibility to observe some overshoot in the step response. Calculated PID controller settings are gathered in Table 4.

All experiments started with the operating point: $Y_G = 50\%$, and $Y_W = 50\%$. In Figs. 12, 13, 14, 15 and 16 the set-point change (ΔSP , $SP_{end} = 50\%$) response signals for control systems with PID and FOPID controllers are compared. Figs. 17, 18, 19 and 20 present response signals to disturbance DI1 for both control systems. Figs. 21, 22, 23 and 24 show the response signals to disturbance DI2 for both control systems. The calculated control quality indices for the control system with the $PI^{\lambda}D^{\mu}$ controller are summarized in Table 5.

7.5 Discussion

The results shown in Tables 3 and 5 prove successful implementation of the $PI^{\lambda}D^{\mu}$ algorithm in the PLC controller. The proposed system evaluated experimentally on a laboratory test stand, allowed tracking of the set-point with high accuracy. It was also robust to induced disturbances (air inlet change, and heat power change).

The first major problem was a $PI^{\lambda}D^{\mu}$ tuning method that requires off-line optimization, and more parameters to choose than a standard PID controller. The tuning rules are based on the Oustaloup continuous approximation. Because of implementation

³ <http://nettoplcsim.sourceforge.net/> (it should be noted that it is necessary to use TIA-Portal V15).

Table 4 PID controller settings (Ziegler-Nichols, $N_D = 8$)

K_{PID}	T_{PID}	$T_{D_{PID}}$
9.768	10.850	2.604

Fig. 12 Response to the set-point change of the system with a PID controller—the process value and the set-point value **Fig. 13** Response to the set-point change of the system with a PID controller—the control signal **Fig. 14** Response to the set-point change of the system with a $PI^\lambda D^\mu$ controller—the process value and the set point value

requirements in the PLC controller, the discrete CFE approximation of fractional operators was used. Therefore, there still exists the problem of interpreting

Fig. 15 Response to the set-point change of the system with a $PI^\lambda D^\mu$ controller—the control signal **Fig. 16** Errors of the system with a PID and a $PI^\lambda D^\mu$ controllers—response to the set-point change **Fig. 17** Response of the system with a PID controller to the disturbance DI1—the process value and the set-point value

optimized controller parameters for a PLC-based discrete $PI^\lambda D^\mu$ control system.

When compared to the SIL simulation, the laboratory implementation was characterized by a four times shorter settling time and no overshoot. It was a surprisingly good result. However, the experimental comparison with the PID controller allows observing other important system properties that are hard to detect in simulations.

Fig. 18 Response of the system with a PID controller to the disturbance DI1—the control value

Fig. 19 Response of the system with a $PI^\lambda D^\mu$ controller to disturbance DI1—the process value and the set-point value, **b** the control value

Fig. 20 Response of the system with a $PI^\lambda D^\mu$ controller to disturbance DI1—the control value

In all evaluated cases, the maximum errors didn't exceed 0.4% (approx. 0.2°C) for $PI^\lambda D^\mu$, and 1.25% (approx. 0.6°C) for PID controller. For the set-point change, an important advantage of the $PI^\lambda D^\mu$ controller was zero overshoot and shorter settling time compared with the PID controller. As it was mentioned before, the occurrence of overshoot for the PID

Fig. 21 Response of the system with a PID controller to disturbance DI2—the process value and the set-point value

Fig. 22 Response of the system with a PID controller to disturbance DI2—the control value

Fig. 23 Response of the system with a $PI^\lambda D^\mu$ controller to disturbance DI2—the process value and the set point value

controller depends on the Ziegler-Nichols tuning method properties.

Both PID and $PI^\lambda D^\mu$ controllers, after proposed parameter tuning, are robust to disturbances. In this case, the advantage of the $PI^\lambda D^\mu$ controller is around two times smaller maximum error comparing to the standard PID controller. However, it has a worse settling time and higher overshoot. Such differences in quality indices for a $PI^\lambda D^\mu$ controller can be described

Fig. 24 Response of the system with a $PI^{\lambda}D^{\mu}$ controller to disturbance DI2—the control value

Table 5 Comparison of the quality indices of the a PID and a $PI^{\lambda}D^{\mu}$ temperature control system

	$e_{\max}$ (%)	e_{st} (%)	γ (%)	T_R (s)
<i>PID</i>				
Set-point change	1.2207	0.0021	9.3685	32.6000
DI1	0.8068	0.0350	10.3899	71.0000
DI2	0.8545	− 0.0027	29.1084	67.0000
<i>FOPID</i>				
Set-point change	0	− 0.0940	0.000	22.6000
DI1	0.3260	− 0.0403	38.7041	90.0000
DI2	0.3518	− 0.0686	31.0784	74.4000

by the visible noise in the control signal (see Figs. 20 and 24). This phenomenon is well known for controllers with derivative action, which amplifies abrupt changes in the error signal (input to the controller). For the PID controller, it also occurred (see Figs. 18 and 22), but the magnitude of noise was smaller. There are many ways to improve signal quality in closed-loop control systems. Methods like noise filtering, feedforward from load disturbances, or process dynamics compensation are used in industrial practice [13, 57]. However, there is no systematic research on such methods for fractional-order control systems.

8 Conclusions

The article describes the design, parameter tuning, and experimental evaluation of a fractional-order $PI^{\lambda}D^{\mu}$ temperature control in a pipeline with induced air-flow

on a laboratory test stand. The $PI^{\lambda}D^{\mu}$ controller was implemented on a standard PLC Siemens S7-1200 controller. Fractional operators were approximated using the Continued Fraction Expansion (CFE). The proposed solution was tested via simulations and experiments on the laboratory test stand, and compared with a standard integer-order PID controller.

The results obtained during the presented research, with the thermal-flow process, showed that to apply the proposed control algorithm it is crucial to develop methods for tuning the controller parameters to minimize the influence of input delays and disturbances. The proposed future work includes research on tuning methods of discrete FOPID controllers with CFE approximation of differ-integral and improvements of signal quality in closed-loop fractional-order control systems.

Funding This research was funded by the European Regional Development Fund in the Research Centre of Advanced Mechatronic Systems project within the Operational Programme Research, Development and Education (Grant Number CZ.02.1.01/0.0/0.0/16_019/0000867). This research was supported by the project SP2020/42, “Development of algorithms and systems for control, measurement and safety applications VI” of Student Grant System, VSB-TU Ostrava.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Åström KJ, Hägglund T (2005) PID controllers: theory, design, and tuning, Second Edition. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709

2. Åström KJ, Wittenmark B (2013) Computer-controlled systems: theory and design, 3rd edn. Courier Corporation, North Chelmsford
3. Azarmi R, Tavakoli-Kakhki M, Sedigh AK, Fatehi A (2015) Analytical design of fractional order PID controllers based on the fractional set-point weighted structure: case study in twin rotor helicopter. *Mechatronics* 31:222–233. <https://doi.org/10.1016/j.mechatronics.2015.08.008>
4. Baranowski J, Bauer W, Zagórska M, Dziwiński T, Piątek P (2015) Time-domain Oustaloup approximation. In: 2015 20th international conference on methods and models in automation and robotics (MMAR), pp 116–120. <https://doi.org/10.1109/MMAR.2015.7283857>
5. Beschi M, Padula F, Visioli A (2017) The generalised iso-damping approach for robust fractional PID controllers design. *Int J Control* 90(6):1157–1164. <https://doi.org/10.1080/00207179.2015.1099076>
6. Caputo M (1967) Linear model of dissipation whose Q is almost frequency independent-II. *Geophys J Int* 13(5):529–539. <https://doi.org/10.1111/j.1365-246X.1967.tb02303.x>
7. Chen YQ, Moore KL (2002) Discretization schemes for fractional-order differentiators and integrators. *IEEE Trans Circuits Syst I Fundam Theory Appl* 49(3):363–367. <https://doi.org/10.1109/81.989172>
8. Dorf RC, Bishop RH (2000) Modern control systems, 9th edn. Prentice-Hall Inc, Englewood Cliffs
9. El-Khazali R (2013) Fractional-order $PI^{\lambda}D^{\mu}$ controller design. *Comput Math Appl* 66(5):639–646. <https://doi.org/10.1016/j.camwa.2013.02.015>
10. Ghasemi S, Tabesh A, Askari-Marnani J (2014) Application of fractional calculus theory to robust controller design for wind turbine generators. *IEEE Trans Energy Convers* 29(3):780–787
11. Gorenflo R, Mainardi F (1997) Fractional calculus: integral and differential equations of fractional order. Springer, Vienna, pp 223–276. https://doi.org/10.1007/978-3-7091-2664-6_5
12. Haber R, Keviczky L (1999) Nonlinear system identification: input-output modeling approach. Springer, Berlin
13. Hagglund T (2013) A unified discussion on signal filtering in PID control. *Control Eng Practice* 21(8):994–1006. <https://doi.org/10.1016/j.conengprac.2013.03.012>
14. Haji V, Monje CA (2017) Fractional order fuzzy-PID control of a combined cycle power plant using particle swarm optimization algorithm with an improved dynamic parameters selection. *Appl Soft Comput* 58:256–264. <https://doi.org/10.1016/j.asoc.2017.04.033>
15. Hassan F, Zolotas A (2017) Impact of fractional order methods on optimized tilt control for rail vehicles. *Fract Calc Appl Anal* 20(3):765–789. <https://doi.org/10.1515/fca-2017-0039>
16. Hilfer R (2000) Applications of fractional calculus in physics. World Scientific, Singapore
17. Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. *Int J Forecast* 22(4):679–688. <https://doi.org/10.1016/j.ijforecast.2006.03.001>
18. Jiangbo Z, Junzheng W (2017) The fractional order pi control for an energy saving electro-hydraulic system. *Trans Inst Meas Control* 39(4):505–519. <https://doi.org/10.1177/0142331215610184>
19. Kaczorek T, Rogowski K (2015) Fractional linear systems and electrical circuits (in Studies in Systems, Decision and Control, vol 13). Springer, Cham. <https://doi.org/10.1007/978-3-319-11361-6>
20. Kolaj W, Możaryn J, Syfert M (2016) PLC-PIDTuner: application for PID tuning with SIMATIC S7 PLC controllers. In: 2016 21st international conference on methods and models in automation and robotics (MMAR), pp 306–311. <https://doi.org/10.1109/MMAR.2016.7575152>
21. Kumar A, Kumar V (2017) Hybridized ABC-GA optimized fractional order fuzzy pre-compensated FOPID control design for 2-DOF robot manipulator. *AEU Int J Electron Commun* 79:219–233. <https://doi.org/10.1016/j.aeue.2017.06.008>
22. Kurek J (2010) Approximation of inertial model with time delay for n-th order inertial systems [in polish]. *Pomiary, Automatyka, Robotyka* 14(11):49–51
23. Lanusse P, Sabatier J (2011) PLC implementation of a CRONE controller. *Fract Calc Appl Anal* 14(4):505–522. <https://doi.org/10.2478/s13540-011-0031-7>
24. Łaskowski M, Wciślik M (2016) Sampling rate impact on the tuning of PID controller parameters. *Int J Electron Telecommun*. <https://doi.org/10.1515/eletel-2016-0005>
25. Laski P (2019) Fractional-order feedback control of a pneumatic servo-drive. *Bull Pol Acad Sci Tech Sci* 67(1):53–59. <https://doi.org/10.24425/bpas.2018.126120>
26. Li C, Zhang N, Lai X, Zhou J, Xu Y (2017) Design of a fractional-order PID controller for a pumped storage unit using a gravitational search algorithm based on the Cauchy and Gaussian mutation. *Inf Sci* 396:162–181. <https://doi.org/10.1016/j.ins.2017.02.026>
27. Lino P, Maione G, Stasi S, Padula F, Visioli A (2017) Synthesis of fractional-order PI controllers and fractional-order filters for industrial electrical drives. *IEEE/CAA J Autom Sin* 4(1):58–69. <https://doi.org/10.1109/JAS.2017.7510325>
28. Liu F, Agrawal OP, Momani S, Leonenko NN, Chen W (2013) Fractional differential equations. *Int J Differ Equ* 2013(1):1–2. <https://doi.org/10.1155/2013/802324>
29. Liu W, Bian GB, Rahman MRU, Zhang H, Chen H, Wu W (2019) Fractional-order PID servo control based on decoupled visual model. *Int J Adapt Control Signal Process* 33(8):1265–1280. <https://doi.org/10.1002/acs.2956>
30. Lorenzo C, Hartley T (2016) The fractional trigonometry. Wiley, New York. <https://doi.org/10.1002/9781119139447.ch1>
31. Ma C, Hori Y (2004) Backlash vibration suppression control of torsional system by novel fractional order PID^k controller. *IEEJ Trans Ind Appl* 124(3):312–317. <https://doi.org/10.1541/ieejias.124.312>
32. Monje C, Chen Y, Vinagre B, Xue D, Feliu-Batlle V (2010) Fractional order systems and control—fundamentals and applications. Springer, London. <https://doi.org/10.1007/978-1-84996-335-0>
33. Monje CA, Vinagre BM, Feliu V, Chen YQ (2008) Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng Pract* 16(7):798–812. <https://doi.org/10.1016/j.conengprac.2007.08.006>
34. Możaryn J, Malinowski K (2014) Tuning rules selection and iterative modification of PID control system parameters. In:

- Březina T, Jabłoński R (eds) *Mechatronics* 2013. Springer International Publishing, Cham, pp 677–684
35. Muresan CI, Folea S, Mois G, Dulf EH (2013) Development and implementation of an FPGA based fractional order controller for a DC motor. *Mechatronics* 23(7):798–804. <https://doi.org/10.1016/j.mechatronics.2013.04.001>
 36. Mystkowski A, Kierdelewicz A (2018) Fractional-order water level control based on PLC: hardware-in-the-loop simulation and experimental validation. *Energies* 11(11):2928. <https://doi.org/10.3390/en11112928>
 37. Nocedal J, Wright SJ (2006) *Numerical Optimization*, 2nd edn. Springer-Verlag, Berlin. <https://doi.org/10.1007/b98874>
 38. O'Dwyer A (2009) *Handbook of PI and PID controller tuning rules*, 3rd edn. World Scientific Publishing CO., Singapore. <https://doi.org/10.1142/p575>
 39. Oldham KB, Spanier J (eds) (1974) *The fractional calculus: theory and applications of differentiation and integration to arbitrary order*. mathematics in science and engineering, vol 111. Elsevier, Amsterdam. [https://doi.org/10.1016/S0076-5392\(09\)60224-1](https://doi.org/10.1016/S0076-5392(09)60224-1)
 40. Oprzędkiewicz K, Mitkowski W, Gawin E (2017) The PLC implementation of fractional-order operator using CFE approximation. *Automation* 550:22–33. https://doi.org/10.1007/978-3-319-54042-9_2
 41. Oprzędkiewicz K (2019) Real-time requirements meeting in PLC control system. *Sci Technol Innov* 5(2):1–8
 42. Oprzędkiewicz K, Gawin T, Gawin E, Mitkowski W (2017) PLC implementations of an elementary fractional order operator. *AUTSYM* 2017
 43. Oprzędkiewicz K, Podsiadło M (2019) The fractional order PID control of the forced air heating system. *Pomiary Automatyka Robotyka* 23(1):5–10. https://doi.org/10.14313/PAR_231/5
 44. Oprzędkiewicz K, Więckowski Ł, Podsiadło M (2020) Discrete, fractional order, cancellation controller. part I: Idea and simulations. In: Szewczyk R, Zieliński C, Kaliczyńska M (eds) *Automation* 2019. Springer International Publishing, Cham, pp 27–38. https://doi.org/10.1007/978-3-030-13273-6_3
 45. Oprzędkiewicz K, Więckowski Ł, Podsiadło M (2020) Discrete, fractional order, cancellation controller. part II: PLC implementation. In: Szewczyk R, Zieliński C, Kaliczyńska M (eds) *Automation* 2019. Springer International Publishing, Cham, pp 39–46. https://doi.org/10.1007/978-3-030-13273-6_4
 46. Oustaloup A, Levron F, Mathieu B, Nanot FM (2000) Frequency-band complex noninteger differentiator: characterization and synthesis. *IEEE Trans Circuits Syst I Fundam Theory Appl* 47(1):25–39. <https://doi.org/10.1109/81.817385>
 47. Oustaloup A, Moreau X, Nouillant M (1996) The CRONE suspension. *Control Eng Pract* 4(8):1101–1108. [https://doi.org/10.1016/0967-0661\(96\)00109-8](https://doi.org/10.1016/0967-0661(96)00109-8)
 48. Ozana S, Docekal T (2016) PID controller design based on global optimization technique with additional constraints. *J Electr Eng* 67(3):160–168. <https://doi.org/10.1515/jee-2016-0023>
 49. Padula F, Visioli A (2011) Tuning rules for optimal PID and fractional-order PID controllers. *J Process Control* 21:69–81. <https://doi.org/10.1016/j.jprocont.2010.10.006>
 50. Pan I, Das S (2012) Chaotic multi-objective optimization based design of fractional order $PI^\lambda D^\mu$ controller in AVR system. *Int J Electr Power Energy Syst* 43(1):393–407. <https://doi.org/10.1016/j.ijepes.2012.06.034>
 51. Petráš I (2009) Realization of fractional order controller based on PLC and its utilization to temperature control. *Transfer inovácií* 14:34–38. <https://doi.org/10.2478/s13540-012-0021-4>
 52. Petráš I (2012) Tuning and implementation methods for fractional-order controllers. *Fract Calc Appl Anal* 15(2):282–303. <https://doi.org/10.2478/s13540-012-0021-4>
 53. Petryszyn J, Możaryn J, Ozana S (2019) Experimental evaluation of PLC based fractional order $PI^\lambda D^\mu$ temperature control in pipeline. In: Awrejcewicz J, Kaźmierczak MJ (eds) *Theoretical approaches in non-linear dynamical systems*, vol 1. Springer International Publishing, Lodz, Poland, pp 419–430
 54. Podlubny I (ed) (1999) *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, mathematics in science and engineering, vol 198. Elsevier, Amsterdam. [https://doi.org/10.1016/S0076-5392\(99\)80026-5](https://doi.org/10.1016/S0076-5392(99)80026-5)
 55. Podlubny I (1999) Fractional-order systems and $PI^\lambda D^\mu$ -controllers. *IEEE Trans Autom Control* 44(1):208–214. <https://doi.org/10.1109/9.739144>
 56. Podlubny I (2002) Geometric and physical interpretation of fractional integral and fractional differentiation. *J Fract Calc Appl Anal* 5(4):367–386
 57. Segovia VR, Hagglund T, Åström K (2014) Measurement noise filtering for common PID tuning rules. *Control Eng Pract* 32:43–63. <https://doi.org/10.1016/j.conengprac.2014.07.005>
 58. Sondhi S, Hote YV (2014) Fractional order PID controller for load frequency control. *Energy Convers Manag* 85:343–353. <https://doi.org/10.1016/j.enconman.2014.05.091>
 59. Valério D, da Costa JS (2006) Tuning of fractional PID controllers with Ziegler–Nichols-type rules. *Sig Process* 86(10):2771–2784. <https://doi.org/10.1016/j.sigpro.2006.02.020>
 60. Wang N, Wang J, Li Z, Tang X, Hou D (2018) Fractional-order PID control strategy on hydraulic-loading system of typical electromechanical platform. *Sensors* 18(9):3024. <https://doi.org/10.3390/s18093024>
 61. Xu Y, Zhou J, Xue X, Fu W, Zhu W, Li C (2016) An adaptively fast fuzzy fractional order PID control for pumped storage hydro unit using improved gravitational search algorithm. *Energy Convers Manag* 111:67–78. <https://doi.org/10.1016/j.enconman.2015.12.049>
 62. Xue D (ed) (2017) *Fractional-order control systems. fundamentals and numerical implementations*. De Gruyter, Berlin
 63. Zafrioui E, Morari M (1986) Design of robust digital controllers and sampling-time selection for SISO systems. *Int J Control* 44(3):711–735. <https://doi.org/10.1080/00207178608933629>
 64. Zeng GQ, Chen J, Dai YX, Li LM, Zheng CW, Chen MR (2015) Design of fractional order PID controller for automatic regulator voltage system based on multi-objective

- extremal optimization. *Neurocomputing* 160:173–184. <https://doi.org/10.1016/j.neucom.2015.02.051>
65. Zhao C, Xue D, Chen Y (2005) A fractional order pid tuning algorithm for a class of fractional order plants. In: *IEEE international conference mechatronics and automation*, 2005, vol 1, pp 216–221. <https://doi.org/10.1109/ICMA.2005.1626550>
66. Zhao J, Wang J, Wang S (2013) Fractional order control to the electro-hydraulic system in insulator fatigue test device. *Mechatronics* 23(7):828–839. <https://doi.org/10.1016/j.mechatronics.2013.02.002>
67. Ziegler J, Nichols N (1993) Optimum setting for automatic controllers (reprint). *J Dyn Syst Meas Contr* 115(2B):220–222. <https://doi.org/10.1115/1.2899060>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.