BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 73(3), 2025, Article number: e153831
DOI: 10.24425/bpasts.2025.153831

CONTROL, INFORMATICS AND ROBOTICS

Digital PID controllers with fractional variable order techniques and two discrete-time operator variants

Dorota MOZYRSKA, Małgorzata WYRWAS*, and Piotr OZIABLO

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract. This paper introduces a control strategy utilizing the fractional variable order PID digital controller (FVOPID). We employ two variants of fractional discrete-time operators with variable order: the Grünwald-Letnikov type and its convolution version. We examine and compare the performance of both controllers types on a higher-order plant. Parameter optimization is conducted using a particle swarm algorithm.

Keywords: fractional variable order operators; PID controllers; discrete-time; optimization.

1. INTRODUCTION

Fractional order calculus extends classical calculus to real-number orders, proving its utility across various scientific and engineering domains such as biophysics, thermodynamics, optimization, and systems modeling [1–4]. Typically, applications assume a constant derivative order. However, exploring time-dependent orders for integrals/summations and derivatives/differences yields intriguing possibilities. While determining the optimal function is complex, a straightforward approach involves piecewise-constant functions, as suggested in [5–9]. This study delves into applying fractional variable-order discrete-time calculus in control theory. Fractional PID controllers, a variant of the classical PID controllers, gained prominence since Podlubny introduced them in 1994 [10, 11]. Subsequent research yielded various designs for fractional PID controllers, aiming for improved robustness against parameter variations in controlled systems [12–16]. Digital fractional controllers offer precise control over processes needing fractional adjustments, boasting reliability, low maintenance, and easy integration into digital systems for remote monitoring and control. The extension to fractional variable-order PID controllers (FVOPID) further enhances adaptability over time [17]. Digital implementations utilize Grñwald-Letnikov operators for FVOPID controllers [5–9].

This paper compares two types of FVOPID controllers applied to two different objects: one is a higher-order system and the second one is an automatic voltage regulation system (AVR). Instead of conventional integration and differentiation, we integrate operators of variable fractional order summation and differences into the classical PID formula. The first version incorporates the fractional variable-order Grünwald-Letnikov operator, termed FVOPID. However, this operator does not take a convolution form. To address this, we propose a discrete-time operator with a convolution form, defining it precisely in subsequent paragraphs. The second controller type, utilizing the convolution operator, is denoted as FVOPID-C. Our aim is to analyze the closed-loop system response with both controller types and check results for two objects. We apply the method to the special objective function and find elements of values of orders and coefficients of controllers using a particle swarm optimization (PSO) method for two different systems. Particle swarm optimization algorithm (PSO) was proposed by J. Kennedy and R. Eberhart in [18, 19]. It is one of the best-known and broadly used methods of solving a variety of optimization problems. It is a population-based algorithm which reflects the intelligence of the swarm. In its earliest form the algorithm was designed to mimic the behavior of a flock of birds searching for food. The algorithm is based on a swarm of individual particles (birds) each of which can be associated with a single potential solution. By the solution we can understand a vector of searched parameters. What is more each particle "remembers" its optimal position and the optimal position of the swarm. In each generation the algorithm updates the velocity and position of the particles until the optimal solution is obtained or the algorithm has reached its maximum number of iterations. A review of PSO algorithms and with the different search approaches and adjustments can be found in [20, 21].

2. FRACTIONAL-VARIABLE-ORDER OPERATORS

In this paper we use two different types of difference fractional-variable-order operators based on the version of the Grünwald-Letnikov definition. The Grünwald-Letnikov variable-order operators can be defined by utilizing oblivion function, the definition of which is presented in the first definition. Let for $h > 0$, $h\mathbb{Z} := \{\ldots, -2h, -h, 0, h, 2h, \ldots\}$.

*e-mail: m.wyrwas@pb.edu.pl

Manuscript submitted 2024-09-19, revised 2025-12-21, initially accepted for publication 2025-02-03, published in May 2025.

Definition 1 [22, 23]**.** Let $k, i \in \mathbb{Z}$ and $\nu : h\mathbb{Z} \to \mathbb{R}_+ \cup \{0\}$ be an order function. Then, an *oblivion function* is defined as

$$a^{\nu(kh)}(i) = \begin{cases} 0, & i < 0 \\ 1, & i = 0 \\ (-1)^i \frac{\nu(kh)(\nu(kh)-1)\cdots(\nu(kh)-i+1)}{i!}, & i > 0 \end{cases}, \quad (1)$$

where $h > 0$ is a sample time.

Formula (1) in Definition 1 can be translated into the recurrence with respect to $k \in \mathbb{N}_0$:

$$\begin{aligned} &a^{\nu(kh)}(0) = 1, \\ &a^{\nu(kh)}(k) = a^{\nu(kh)}(k-1)\left(1 - \frac{\nu(kh)+1}{k}\right), \quad k \geq 1. \end{aligned} \quad (2)$$

The recurrent form of the oblivion function given by (2) allows it to be calculated more efficiently than by using equation (1), since there are fewer operations performed on smaller numbers needed, particularly if values of the order function remain constant for some periods of time or when it gets repeated. One of the properties of the oblivion function is that for positive order values the sequence of $a^{\nu(kh)}(k)$ is quickly tending to zero. More information regarding the oblivion function and its properties can be found in [22, 24, 25].

Let us present two types of fractional-variable-order differences. The first one is the standard fractional-variable-order Grünwald-Letnikov difference (FVOGLD), which corresponds to known in the literature Type $\mathscr{A}$ operator.

Definition 2. (FVOGLD) [[23]] Let $x : h\mathbb{Z} \to \mathbb{R}$ be a bounded function. The fractional-variable-order Grünwald-Letnikov difference operator of function x with step $h > 0$ and an order function $\nu : h\mathbb{Z} \to \mathbb{R}$ started at $t = 0$ is defined as a finite sum

$$\Delta_h^{\nu(kh)} x(kh) = \sum_{i=0}^{k} h^{-\nu(kh)} a^{\nu(kh)}(i) x(kh - ih)$$
$$= \begin{bmatrix} 1 & a^{\nu(kh)}(1) & \cdots & a^{\nu(kh)}(k) \end{bmatrix} \begin{bmatrix} x(kh) \\ x(kh-h) \\ \cdots \\ x(h) \\ x(0) \end{bmatrix} h^{-\nu(kh)}. \quad (3)$$

When the sample time h goes to 0, Definition 2, which represents fractional-variable-order difference, can be treated as a discrete-time version of fractional-variable-order derivative. It is worth noting that for constant order function $\nu(kh) \equiv \alpha$ the operator presented in Definition 2 coincides with the Grünwald-Letnikov fractional-order backward difference. Moreover, for the negative order values ($\alpha < 0$) it becomes fractional-order summation (which in continuous time system corresponds to integral action). The definition agrees with the Type $\mathscr{A}$ operator presented in [5, 6].

The second definition which is named fractional-variable-order Grünwald-Letnikov difference of convolution type (FVOGLD-C) was introduced in [26] and is presented in the next definition.

Definition 3 [26]**.** Let $x : h\mathbb{Z} \to \mathbb{R}$ be a bounded function. The fractional-variable-order Grünwald-Letnikov difference–convolution type operator (FVOGLD-C) of function x with step $h > 0$ and an order function $\nu : h\mathbb{Z} \to \mathbb{R}$ is defined as a finite sum

$$\Delta_h^{\nu(\cdot)} x(kh) = \sum_{i=0}^{k} h^{-\nu(ih)} a^{\nu(ih)}(i) x(kh - ih)$$
$$= \begin{bmatrix} \frac{1}{h^{\nu(0)}} & \frac{a^{\nu(h)}(1)}{h^{\nu(h)}} & \cdots & \frac{a^{\nu(kh)}(k)}{h^{\nu(kh)}} \end{bmatrix} \begin{bmatrix} x(kh) \\ x(kh-h) \\ \cdots \\ x(h) \\ x(0) \end{bmatrix}. \quad (4)$$

As we can see FVOGLD-C is a discrete convolution which can be represented by:

$$\Delta_h^{\nu(\cdot)} x(kh) = (\mathfrak{a} * \overline{x})(k) = (\overline{x} * \mathfrak{a})(k), \quad (5)$$

where "$*$" denotes the convolution operator, $\mathfrak{a}(i) := h^{-\nu(ih)} a^{\nu(ih)}(i)$ and $\overline{x}(k) := x(kh)$. The advantage of the convolution operator is that one can compute the image of this operator in $\mathscr{Z}$-transform, i.e.,

$$\mathscr{Z}\left[\Delta_h^{\nu(\cdot)} x\right](z) = X(z)\mathscr{A}(z), \quad (6)$$

where $X(z) := \mathscr{Z}[\overline{x}](z)$ and

$$\mathscr{A}(z) := \sum_{i=0}^{\infty} (-1)^i \binom{-\nu(ih)}{i} z^{-i} h^{-\nu(ih)}.$$

The transform function gives a possibility of stability analysis of the system described by the presented convolution operator which (at least for now) is not possible for the fractional-variable-order Grünwald-Letnikov difference of Type $\mathscr{A}$. Both presented operators are evaluated by multiplying the vector of oblivion function values by the vector of x function values. They have a property called "forgetting" the old values and this is where the name "oblivion function" came from.

3. METHODS

FVOPID controller can be defined in a similar way to a classical PID or FOPID. In this case the constant order summation and difference terms are replaced by the variable-order counterparts like e.g., the Grünwald-Letnikov operators, which for positive order function values realize the operation of difference (derivative in continuous-time domain), while for negative order function values are used to calculate the summation (integral) term of the controller. The general structure of discrete-time FVOPID controller is presented in Fig. 1, where $h > 0$ is the controller sample time, $k \in \mathbb{N}$ is the number of a sample, μ is the order function of summation term, ν is the order function of difference term, $\Delta_h^{-\mu}$ and Δ_h^{ν} represent fractional-variable-order (FVO) discrete-time operators and K_p, K_i, K_d are the parameters that are also common for PID controllers.

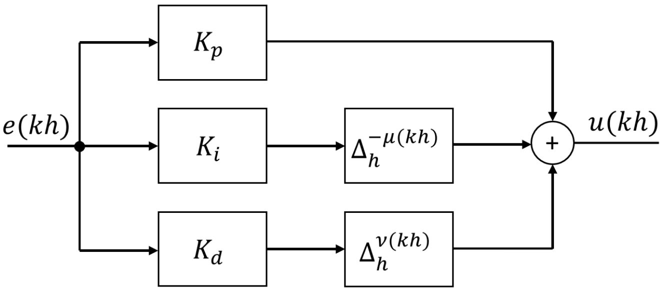

Fig. 1. FVOPID controller schema

FVOPID controllers can utilize different types of FVO operators. This work mainly focuses on the discrete-time Grünwald-Letnikov FVOGLD operator and the Grünwald-Letnikov operator of convolution type (FVOGLD-C), the calculation of which are based on the previously presented Definition 2 and 3, respectively.

The equation of FVOPID controller of Type $\mathscr{A}$ is as follows:

$$u(kh) = K_p e(kh) + K_i \Delta_h^{\mu(kh)} e(kh) + K_d \Delta_h^{\nu(kh)} e(kh) \tag{7}$$

and the equation of FVOPID controller of convolution type is given by

$$u(kh) = K_p e(kh) + K_i \Delta_h^{\mu(\cdot)} e(kh) + K_d \Delta_h^{\nu(\cdot)} e(kh). \tag{8}$$

The orders μ and ν are calculated as follows:

$$\mu(kh) = \begin{cases} \mu_1, & \text{for } \rho(kh) > \rho_1 \\ \mu_2, & \text{for } \rho(kh) \in (\rho_2, \rho_1\rangle \\ \vdots & \\ \mu_{n-1}, & \text{for } \rho(kh) \in (\rho_n, \rho_{n-1}\rangle \\ \mu_n, & \text{for } \rho(kh) \leqslant \rho_{n-1} \end{cases}, \tag{9}$$

$$\nu(kh) = \begin{cases} \nu_1, & \text{for } \rho(kh) > \rho_1 \\ \nu_2, & \text{for } \rho(kh) \in (\rho_2, \rho_1\rangle \\ \vdots & \\ \nu_{n-1}, & \text{for } \rho(kh) \in (\rho_{n-1}, \rho_{n-2}\rangle \\ \nu_n, & \text{for } \rho(kh) \leqslant \rho_{n-1} \end{cases} \tag{10}$$

and $\rho(kh)$ is given by

$$\rho(kh) := \frac{e(kh)}{r(kh)}, \tag{11}$$

where $e(kh)$ is the error value and $r(kh)$ is the setpoint value at time kh.

Presented controllers (7) and (8) meet the following assumptions:

1. The designed controller is discrete-time with sampling time $h = 0.002$ [s].
2. The simulation time for the parameters optimization process is set to 2 [s].
3. The controller summation order function μ and difference order function ν are given by the equations (9) and (10), respectively.
4. Both order functions μ and ν have the same number of possible values (equal to 3, 5 or 7) and the values of the functions are changed at the same time, as represented by (9) and (10).
5. Values of the order functions are the parameters of the controller which are searched by the optimization algorithms.
6. Additional parameters of the controller searched by the optimization algorithms are: K_p, K_i, K_d (see equations (7) and (8)).
7. The number of possible order function values and the values of $\rho_1 - \rho_{n-1}$ from the equations (9) and (10) are the controller architecture specific constants and are not adjusted by the optimization algorithms.

Particle Swarm optimization (PSO) algorithm is set to minimize multi-criteria objective function the general form of which is given by the following equation:

$$OF = w_1 \sum_{k=0}^{N} |e(kh)| kh^2 + w_2 OS + \frac{w_3}{500} \sum_{k=N-499}^{N} |e(kh)| + w_4 t_s, \tag{12}$$

where h is a sampling time, $e(kh)$ is a control error, OS is an overshoot, t_s is a settling time and N is the total number of steps. If during the parameters optimization process the settling time is greater than the simulation time (which means that it is 'Not a Number' – NaN value in MATLAB), then t_s is assigned with 3 – a value that is greater than the simulation time, which allows the objective function value to be determined for the given case (instead of assigning it with NaN). The weighting coefficients of the objective function are set as $w_1 = 1$, $w_2 = 0.2$, $w_3 = 100$, $w_4 = 5$. The coefficient w_3 in the presented equation is multiplied by the average absolute value of the last 500 samples of control error, which can be associated to the steady state error.

PSO algorithm was executed 30 times for PID, FOPID, FVOPID and FVOPID-C controllers, where the number of particles was set to:

- 30 for PID controller,
- 100 for FOPID controller,
- 200 for FVOPID and FVOPID-C controllers.

Searched parameters of the controllers were set with the following boundaries:

- K_p, K_i, K_d – from 0 to 30,
- orders – from 0.5 to 2.

The simulations were performed for variable order controllers with 3, 5 and 7 different values of summation and difference orders, with the order functions given by (9) and (10). The order functions of variable-order controllers with 3 different order values are as follows:

$$\mu(kh) = \begin{cases} \mu_1, & \text{for } \rho(kh) > 0.66 \\ \mu_2, & \text{for } \rho(kh) \in (0.33, 0.66\rangle \\ \mu_3, & \text{for } \rho(kh) \leq 0.33 \end{cases}, \tag{13}$$

$$\nu(kh) = \begin{cases} \nu_1, & \text{for } \rho(kh) > 0.66 \\ \nu_2, & \text{for } \rho(kh) \in (0.33, 0.66\rangle \\ \nu_3, & \text{for } \rho(kh) \leqslant 0.33 \end{cases} . \tag{14}$$

The order functions of variable-order controllers with 5 different values of the order function are given by:

$$\mu(kh) = \begin{cases} \mu_1, & \text{for } \rho(kh) > 0.8 \\ \mu_2, & \text{for } \rho(kh) \in (0.6, 0.8\rangle \\ \mu_3, & \text{for } \rho(kh) \in (0.4, 0.6\rangle \\ \mu_4, & \text{for } \rho(kh) \in (0.2, 0.4\rangle \\ \mu_5, & \text{for } \rho(kh) \leqslant 0.2 \end{cases} , \tag{15}$$

$$\nu(kh) = \begin{cases} \nu_1, & \text{for } \rho(kh) > 0.8 \\ \nu_2, & \text{for } \rho(kh) \in (0.6, 0.8\rangle \\ \nu_3, & \text{for } \rho(kh) \in (0.4, 0.6\rangle \\ \nu_4, & \text{for } \rho(kh) \in (0.2, 0.4\rangle \\ \nu_5, & \text{for } \rho(kh) \leqslant 0.2 \end{cases} . \tag{16}$$

The order functions of variable-order controllers with 7 different values of the order function are as follows:

$$\mu(kh) = \begin{cases} \mu_1, & \text{for } \rho(kh) > 0.84 \\ \mu_2, & \text{for } \rho(kh) \in (0.7, 0.84\rangle \\ \mu_3, & \text{for } \rho(kh) \in (0.56, 0.7\rangle \\ \mu_4, & \text{for } \rho(kh) \in (0.42, 0.56\rangle \\ \mu_5, & \text{for } \rho(kh) \in (0.28, 0.42\rangle \\ \mu_6, & \text{for } \rho(kh) \in (0.14, 0.4\rangle \\ \mu_7, & \text{for } \rho(kh) \leqslant 0.14 \end{cases} , \tag{17}$$

$$\nu(kh) = \begin{cases} \nu_1, & \text{for } \rho(kh) > 0.84 \\ \nu_2, & \text{for } \rho(kh) \in (0.7, 0.84\rangle \\ \nu_3, & \text{for } \rho(kh) \in (0.56, 0.7\rangle \\ \nu_4, & \text{for } \rho(kh) \in (0.42, 0.56\rangle \\ \nu_5, & \text{for } \rho(kh) \in (0.28, 0.42\rangle \\ \nu_6, & \text{for } \rho(kh) \in (0.14, 0.28\rangle \\ \nu_7, & \text{for } \rho(kh) \leqslant 0.14 \end{cases} . \tag{18}$$

4. RESULTS

4.1. Results for higher-order plant

The first plant object that we consider for making a comparison between two types of considered controllers is a higher-order system given by the following equation

$$G(s) = \frac{2(15s+1)}{(20s+1)(s+1)(0.1s+1)^2} . \tag{19}$$

The results of 30 algorithm executions are presented in Table 1 (the number of order of variable-order controllers is shown in brackets). The parameters and qualitative criteria of optimal PID, FOPID, FVOPID and FVOPID-C (5 order values) controllers are presented in Table 2 (qualitative criteria like, e.g., rise time and overshoot were calculated with MATLAB *stepinfo* function). The parameters and qualitative criteria of FVOPID and FVOPID-C for the controllers of 3 and 7 orders are presented in Table 3.

Step responses of the best (in terms of the objective function) controllers of each type are presented in Figs. 2–4.

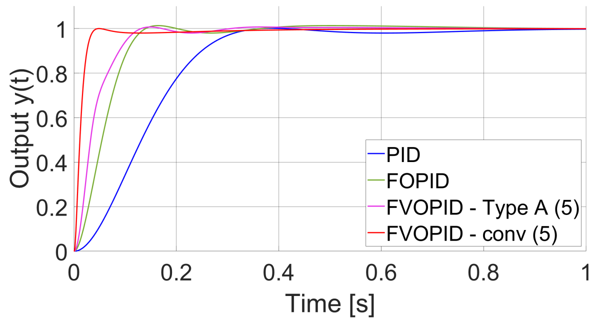

Fig. 2. Step response comparison for higher-order system (19)

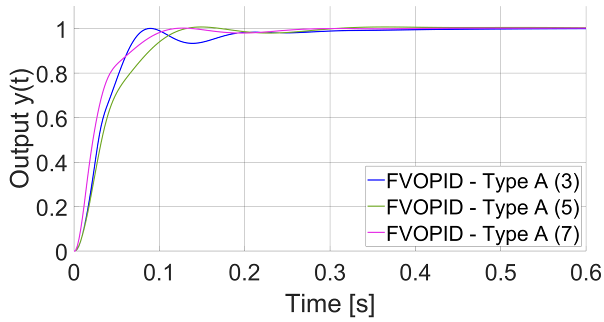

Fig. 3. Step response comparison of FVOPID controllers with 3, 5 and 7 orders for higher-order system (19)

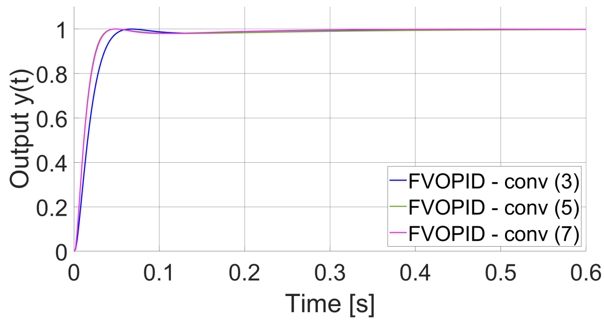

Fig. 4. Step response comparison of FVOPID-C controllers with 3, 5 and 7 orders (almost the same) for higher-order system (19)

4.2. Results for automatic voltage regulation system

The architecture of the controllers presented in this work was also described in detail in the Ph.D. dissertation [27], where the author used yellow saddle goatfish algorithm (YSGA) and particle swarm optimization (PSO) to find the optimal controller parameters for automatic voltage regulation (AVR) system, whose schema is presented in Fig. 5.

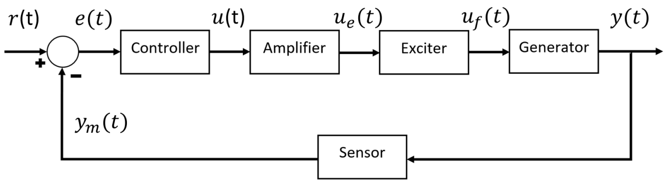

Fig. 5. AVR closed loop schema [27]

Every block of the presented system can be described by the first-order transfer function with gain K and time constant τ, given by the following general equation

$$G(s) = \frac{K}{1+\tau s} \tag{20}$$

with the parameters presented in Table 4. The author of [27] used YSGA algorithm to find the parameters of FVOPID and FVOPID-C controllers and PSO algorithm to tune FVOPID. The optimization algorithms were in this case set to minimize a similar objective function as the one presented in the current work. Additionally, the simulation setup was very close to the one presented in the current work with two differences: (1) the simulation time for the parameters optimization process was set to $3[s]$; (2) The coefficient w_3 of the objective function, associated with the steady state error, was multiplied by the average of the absolute value of the last 750 samples.

The other aspects, such as the sampling time equal to 0.002 $[s]$ or setting the settling time value during the optimization process to 3 (in this case it is the value of settling time equal to the time of the simulation) when the settling time is NaN, were the same.

Table 1

Value of the objective function for 30 PSO algorithm executions – higher-order system (19)

Type of controller	Mean	Min.	Max.	Stdandard deviation	OF eval. number
PID	1.70099959	1.70005226	1.71305749	0.00233245	6 033
FOPID	1.00386171	0.99702797	1.01130845	0.00394678	17 493
FVOPID (3)	1.37267525	1.01576458	1.82054353	0.28664918	66 993
FVOPID (5)	1.01799259	0.76701616	1.3149858	0.15999803	68 300
FVOPID (7)	0.98483434	0.57160884	1.39299451	0.21296776	65 093
FVOPID-C (3)	0.46297288	0.25030152	1.12340814	0.19697810	73 213
FVOPID-C (5)	0.42771446	0.20439328	0.68164722	0.12692378	77 713
FVOPID-C (7)	0.45877611	0.24256997	0.81513004	0.1219751	57 720

Table 2

Parameters and qualitative criteria of controllers tuned with PSO algorithm – higher-order system (19)

	PID	FOPID	FVOPID(5)	FVOPID-C(5)
K_p	5.230361	15.345238	29.770907	7.375620
K_i	4.347479	10.334137	13.587000	29.999129
K_d	0.770167	1.336096	3.424964	1.597124
μ_1	1	1.100071	1.459632	1.378726
μ_2	1	1.100071	0.501347	1.498402
μ_3	1	1.100071	0.548063	1.430928
μ_4	1	1.100071	1.239560	1.892745
μ_5	1	1.100071	1.122641	1.046320
ν_1	1	1.268984	1.24127	1.734221
ν_2	1	1.268984	1.412751	1.856177
ν_3	1	1.268984	1.645277	1.827972
ν_4	1	1.268984	1.648006	1.806695
ν_5	1	1.268984	1.169249	1.217366
Rise time [s]	0.20256	0.08627	0.07787	0.02148
Settling time [s]	0.65778	0.28700	0.27101	0.13419
Overshoot [%]	0	1.196	0.344	0.007
OF value	1.70005226	0.99702797	0.76701616	0.20439328

Table 3
Parameters and qualitative criteria of controllers tuned with PSO algorithm – higher-order system (19)

	FVOPID(3)	FVOPID(7)	FVOPID-C(3)	FVOPID-C(7)
K_p	2.858454	16.916252	8.049215	29.835024
K_i	11.231374	17.645302	26.708781	10.195264
K_d	7.600340	5.105802	1.505233	0.312751
μ_1	1.996930	0.738103	0.766660	1.067156
μ_2	1.980182	1.322293	1.542557	1.529578
μ_3	1.392513	1.180508	0.985083	0.536497
μ_4	–	–	1.314557	0.500000
μ_5	–	–	1.190146	1.547862
μ_6	–	–	1.482524	1.766692
μ_7	–	–	1.158861	0.969161
ν_1	1.097044	1.307180	1.662227	1.767538
ν_2	1.705756	1.608851	1.824524	1.571075
ν_3	1.231615	0.866664	1.985346	1.610845
ν_4	–	0.719831	–	1.994750
ν_5	–	1.054925	–	0.834396
ν_6	–	1.582877	–	0.637365
ν_7	–	1.156963	–	0.674277
Rise time [s]	0.05186	0.06018	0.03071	0.02202
Settling time [s]	0.24816	0.20825	0.04959	0.11719
Overshoot [%]	0.008	0.108	0	0
OF value	1.01576458	0.57160884	0.25030152	0.24256997

Table 4
Transfer function parameters of AVR system components, see [28, 29]

Component	Gain K value	Time constant τ value
Amplifier	10	0.1
Exciter	1	0.4
Generator	1	1
Sensor	1	0.01

Results presented in [27] for AVR system with FVOPID controllers tuned with PSO are consistent with the results presented in the current work. For more details about FVOPID controllers tuned by PSO algorithm for AVR system, refer to [27]. When it comes to FVOPID-C controllers for the AVR system, if we tune them with PSO algorithm using the objective function presented in [27] and for the following boundaries assigned to searched parameters: K_p – from 0.5 to 2, K_i – from 0.1 to 2, K_d – from 0.1 to 1, orders – from 0.5 to 2, we get the controllers whose parameters and qualitative criteria are presented in Table 6. More information about the values of the objective function obtained during the PSO optimization is shown in Table 5. Figure 6 presents step

Table 5
Value of the objective function for 30 PSO algorithm executions – AVR system

Type of controller	Mean	Min.	Max.	Stdandard deviation	OF eval. number
FVOPID-C (3)	0.75519356	0.67199408	0.90442869	0.19697810	70 693
FVOPID-C (5)	0.78080381	0.47511343	1.32523547	0.22973807	77 173
FVOPID-C (7)	0.89983813	0.54265053	1.75317238	0.29508651	87 000

Table 6
Parameters and qualitative criteria of FVOPID-C controllers tuned by PSO algorithm – AVR system

	FVOPID-C(3)	FVOPID-C(5)	FVOPID-C(7)
K_p	1.899984	0.789087	1.795860
K_i	1.526855	2.000000	1.925191
K_d	0.380251	0.406382	0.376431
μ_1	0.500188	1.319649	1.373422
μ_2	0.703487	1.976036	1.898783
μ_3	1.002305	0.884773	1.663491
μ_4	–	0.547920	1.968954
μ_5	–	0.991608	1.396238
μ_6	–	–	1.080135
μ_7	–	–	1.000431
ν_1	1.520522	1.629138	1.601490
ν_2	1.804728	1.775237	1.838463
ν_3	1.989754	1.792022	1.393560
ν_4	–	1.778852	1.800105
ν_5	–	0.985082	1.764865
ν_6	–	–	1.506978
ν_7	–	–	1.005568
Rise time [s]	0.07813	0.05198	0.06178
Settling time [s]	0.12056	0.18814	0.09428
Overshoot [%]	0.125	0.441	0.217
OF value	0.67199408	0.47511343	0.54265053

response of AVR system with PID, FOPID and FVOPID controllers described in [27] and FVOPID-C controller (all tuned with PSO algorithm).

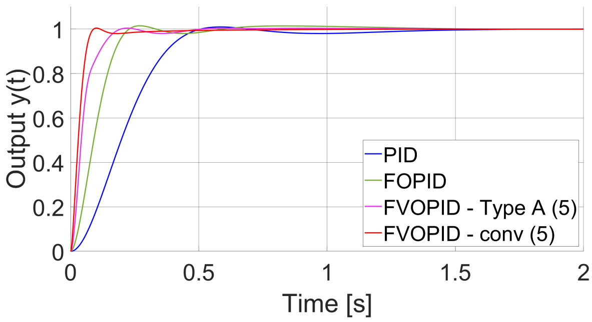

Fig. 6. Step response comparison for AVR system

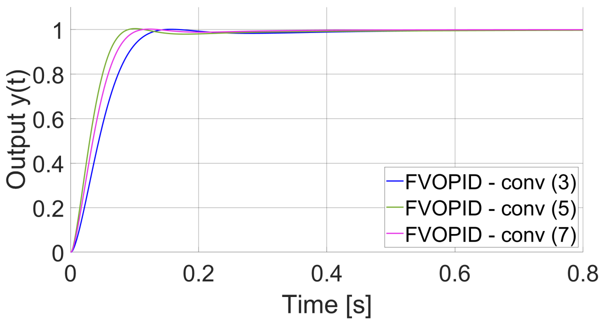

Fig. 7. Step response comparison of FVOPID convolution controllers with 3, 5 and 7 orders for AVR system

In Fig. 7 we can see a comparison of step responses generated by FVOPID-C controller for different numbers of orders.

5. CONCLUSIONS AND FUTURE WORK

5.1. Conclusions

1. Variable order controllers obtained the lowest values of the objective function.
2. The best values of the objective functions for both objects and various numbers of piecewise-constant orders were obtained by FVOPID-C.
3. The lowest rise time (time required by the response to rise from 10% to 90% of its final value) was obtained by FVOPID-C.
4. FVOPID-C controller generally gives better results than the FVOPID, FOPID and PID for both considered types of objects. Moreover, it allows the analysis to be extended to the transform methods due to its convolution form.
5. FVOPID-C controller gives better results with lower number of piece-wise constant orders than FVOPID.

5.2. Future works

Variable-order controllers represent a promising advancement in control systems, offering greater flexibility and adaptability compared to traditional PID controllers. These controllers introduce concepts of fractional calculus, allowing for non-integer order differentiation and integration. FVOPID controllers have the potential to enhance control performance, particularly in systems with complex dynamics or time-varying characteristics. Future research could focus on developing algorithms and methodologies to exploit the flexibility of fractional orders to improve control performance in various industrial applications. Additionally, future research could explore the application of fractional variable order PID controllers in nonlinear control systems, such as chaotic systems, biological systems, or systems with hysteresis, to address challenges that conventional integer-order controllers may encounter.

ACKNOWLEDGEMENTS

The work of D. Mozyrska and M. Wyrwas was supported by Bialystok University of Technology grant No. WZ/WI-IIT/2/2023 and funded by the resources for research by the Ministry of Science and Higher Education.

REFERENCES

[1] R. Hilfer, *Applications of Fractional Calculus in Physics*. Singapore: World Scientific Publishing Company, 2006.

[2] I.M. Batiha, S. Momani, A. Ouannas, Z. Momani, and S.B. Hadid, "Fractional-order COVID-19 pandemic outbreak: Modeling and stability analysis," *Int. J. Biomath.*, vol. 15, no. 01, p. 2150090, 2022, doi: 10.1142/S179352452150090X.

[3] S. Momani, N. Djeddi, M. Al-Smadi, and S. Al-Omari, "Numerical investigation for Caputo-Fabrizio fractional Riccati and Bernoulli equations using iterative reproducing kernel method," *Appl. Num. Math.*, vol. 170, pp. 418–434, 2021, doi: 10.1016/j.apnum.2021.08.005.

[4] M. Al-Smadi, "Fractional residual series for conformable time-fractional Sawada-Kotera-Ito, Lax, and Kaup-Kupershmidt equations of seventh order," *Mathematical Methods in the Applied Sciences*, pp. 1–22, 2021, doi: 10.1002/mma.7507.

[5] D. Sierociuk and W. Malesza, "Fractional variable order anti-windup control strategy," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 66, pp. 427–432, 2018, doi: 10.24425/124258.

[6] D. Sierociuk, W. Malesza, and M. Macias, "On a new definition of fractional variable-order derivative," *Proc. of the 14th International Carpathian Control Conference (ICCC), Rytro, Poland*, pp. 340–345, 2013.

[7] D. Valerio and J. da Costa, "Variable-order fractional derivatives and their numerical approximations," *Signal Process.*, vol. 91, pp. 470–483, 2011.

[8] P. Oziablo, D. Mozyrska, and M. Wyrwas, "Discrete-time fractional, variable-order pid controller for a plant with delay," *Entropy*, vol. 22, no. 7, p. 771, 2020, doi: 10.3390/e22070771.

[9] P. Oziablo, D. Mozyrska, and M. Wyrwas, "A digital PID controller based on Grünwald-Letnikov fractional-, variable-order operator," in *24th International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje*, 2019, pp. 460–465.

[10] I. Podlubny, *Fractional-order systems and fractional-order controllers*. Slovak Academy of Sciences Institute of Experimental Physics, 1994.

[11] I. Podlubny, "Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers," *IEEE Trans. Autom. Control*, vol. 44, pp. 208–214, 1999.

[12] P. Shaha and S. Agasheb, "Review of fractional pid controller." *Mechatronics*, vol. 38, pp. 29–41, 2016.

[13] Y. Chen, "Ubiquitous fractional order controls?" *IFAC Proc. Vol.*, vol. 39, no. 11, pp. 481–492, 2006.

[14] C. Monje, Y. Chen, B. Vinagre, and V. Feliubatlle, *Fractional-Order Systems and Fractional-Order Controllers*. Springer Science & Business Media, 2010.

[15] I. Petras, "Tuning and implementation methods for fractional-order controllers." *Fract. Calc. Appl. Anal.*, vol. 15, no. 2, pp. 282–303, 2012.

[16] S. Debarma, L. Saikia, and N. Sinha, "Automatic generation control using two degree of freedom fractional order PID controller." *Int. J. Electr. Power Energy Syst.*, vol. 58, pp. 120–129, 2014.

[17] P. Ostalczyk, "Variable-, fractional-order discrete pid controllers," in *2012 17th International Conference on Methods Models in Automation Robotics (MMAR)*, 2012, pp. 534–539, doi: 10.1109/MMAR.2012.6347829.

[18] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proc. Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43, doi: 10.1109/MHS.1995.494215.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[20] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Comput.*, vol. 22, p. 387–408, 2018, doi: 10.1007/s00500-016-2474-6.

[21] M. El-Shorbagy and A.E. Hassanien, "Particle swarm optimization from theory to applications," *Int. J. Rough Sets Data Anal.*, vol. 5, p. 24, 01 2018, doi: 10.4018/IJRSDA.2018040101.

[22] P. Ostalczyk, "Stability analysis of a discrete-time system with a variable-, fractional-order controller," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 58, pp. 613–619, 2010.

[23] D. Mozyrska and P. Ostalczyk, "Generalized fractional-order discrete-time integrator," *Complexity*, p. 3452409, 2017.

[24] P. Ostalczyk, *Zarys rachunku różniczkowo-całkowego ułamkowych rzędów*. Wydawnictwo Politechniki Łódzkiej, 2008.

[25] D. Mozyrska and P. Ostalczyk, "Variable-fractional-order Grünwald-Letnikov backward difference selected properties," *Proc. 39th International Conference on Telecommunications and Signal Processing*, 2016.

[26] D. Mozyrska and M. Wyrwas, "Systems with fractional variable-order difference operator of convolution type and its stability," *Elektron. Elektrotech.*, vol. 24, no. 5, pp. 69–73, 2018.

[27] P. Oziablo, *Fractional-variable-order digital PID controllers based on metaheuristic algorithms optimization*. PhD dissertation, Bialystok University of Technology, 2023.

[28] M. Micev, M. Ćalasan, and D. Oliva, "Fractional order PID controller design for an AVR system using chaotic Yellow Saddle Goatfish algorithm," *Mathematics*, vol. 8, no. 7, 2020, doi: 10.3390/math8071182.

[29] A. Sikander, P. Thakur, R. Bansal, and S. Rajasekar, "A novel technique to design cuckoo search based FOPID controller for AVR in power systems," *Comput. Electr. Eng.*, vol. 70, p. 261–274, 2018.