Chaos, Solitons and Fractals 213 (2026) 119217

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos

A nested weighted framework for Podlubny-type fractional PID controllers

Antonio Di Teodoro [a], Juan J. Gude [a,*], Oscar Camacho [b]

[a] Faculty of Engineering, University of Deusto, Avda. de la Universidades, 24, 48007 Bilbao, Spain
[b] Colegio de Ciencias e Ingenierías, Universidad San Francisco de Quito, Diego de Robles s/n, vía Interoceánica, 170901 Quito, Ecuador

ARTICLE INFO

Keywords:
Fractional calculus
Generalized fractional operators
Nested weighted operators
Fractional-order control
Podlubny-type fractional
PID controllers
Controller synthesis

ABSTRACT

This paper presents a unified nested weighted framework for the systematic synthesis of Podlubny-type fractional PID controllers through the constructive generation of generalized fractional operators. The proposed methodology adopts a composition-based perspective in which generalized fractional operators are recursively constructed from weighted differential and integral mappings, allowing classical formulations, including the Riemann–Liouville and Caputo operators, to be recovered as particular cases. Within this framework, a novel pulling-type weighted fractional operator is introduced and its fundamental analytical properties, including linearity, kernel characterization, boundedness, semigroup behavior, and Laplace-domain representation, are rigorously established. These results provide the mathematical foundation for deriving a generalized family of weighted Podlubny-type fractional PID controllers, where the weighting functions become explicit design parameters governing memory distribution and closed-loop dynamics. Consequently, the proposed framework transforms the construction of generalized fractional operators into a systematic methodology for fractional controller synthesis, establishing a constructive link between generalized fractional calculus and fractional-order control design.

1. Introduction

During the last three decades, fractional-order control has evolved from a primarily theoretical research topic into a mature methodology for addressing the complex dynamics exhibited by many practical systems. Unlike classical integer-order formulations, fractional operators naturally describe memory and hereditary effects, providing more accurate representations of diffusion phenomena, viscoelastic materials, thermal processes, electrochemical systems, and distributed-parameter dynamics [1–3]. Consequently, fractional-order controllers have demonstrated significant improvements in robustness, disturbance rejection, and tracking performance across a broad range of engineering applications, including robotics, chemical processes, energy systems, biomedical engineering, and industrial automation [4,5]. Despite these advances, most existing controller architectures continue to rely on a relatively limited set of predefined fractional operators, restricting the structural flexibility available for developing new classes of control laws with tailored memory characteristics.

Among the various fractional-order control strategies, the fractional-order PID (FOPID) controller proposed by Podlubny remains the most influential and widely adopted architecture [1]. By introducing independent fractional integration and differentiation orders, this formulation extends the classical PID controller from three to five tuning parameters, providing additional flexibility for shaping closed-loop dynamics. Since its original introduction, extensive research has focused on tuning methodologies, optimization

* Corresponding author.
E-mail addresses: a.diteodoro@deusto.es (A. Di Teodoro), jgude@deusto.es (J.J. Gude), ocamacho@usfq.edu.ec (O. Camacho).

https://doi.org/10.1016/j.chaos.2026.119217
Received 11 July 2026; Received in revised form 4 September 2026; Accepted 14 September 2026
Available online 22 September 2026
0960-0779/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

algorithms, digital implementation, robustness analysis, and practical applications [2,4,6–9]. Recent experimental studies have further demonstrated the applicability of fractional-order control to practical process systems. Continuous and discrete FOPID controllers have been implemented in real time on coupled-tank systems [10], while optimized discrete FOPID control has been applied to pressure swing adsorption processes for bioethanol production under reference changes and external disturbances [11]. More recently, fractional-order integral state-feedback control has also been investigated for disturbance attenuation in pressure swing adsorption systems [12]. Beyond these experimental developments, adaptive, variable-order, intelligent, and data-driven formulations have further expanded the applicability of the original architecture [5,13]. Weighted FOPID structures have also recently been explored for the control of nonlinear systems with variable time delay [14]. Nevertheless, these developments have generally focused on specific controller formulations, tuning, and implementation, rather than on providing a systematic framework for extending the underlying fractional operator and deriving broader families of controller architectures.

Meanwhile, fractional calculus has experienced remarkable theoretical advances through the development of generalized operators, including weighted, proportional, tempered, and non-singular kernel formulations, considerably extending the mathematical framework beyond the classical Riemann–Liouville and Caputo definitions [15–17]. Many of these operators are constructed through the systematic composition of elementary fractional mappings, leading to operators with customized memory characteristics [18]. Although these developments have substantially enriched the theory of fractional calculus, they have had only limited impact on controller synthesis. Consequently, the evolution of generalized fractional operators and that of fractional PID controllers have largely followed independent paths, leaving a significant gap between modern operator theory and fractional control design. As a result, no systematic methodology currently exists for translating generalized fractional operator theory into generalized controller architectures.

Motivated by this gap, this paper develops a unified composition-based framework for constructing nested weighted fractional operators and exploiting them in the systematic synthesis of Podlubny-type fractional PID controllers. Rather than proposing another tuning strategy or an isolated FOPID variant, the proposed methodology enlarges the controller design space by generalizing the fractional operator itself through recursive weighted compositions, where the weighting functions become structural design variables. Within this framework, the classical $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller naturally emerges as a particular case, while broader families of weighted FOPID controllers can be systematically generated through different nested operator compositions.

To establish this framework, a novel pulling-type weighted fractional operator is first introduced and its main analytical properties, including linearity, kernel characterization, boundedness, and semigroup behavior, are rigorously established. These analytical results provide the mathematical basis for deriving a generalized family of weighted Podlubny-type fractional PID controllers. The resulting controller synthesis methodology is finally validated through real experimental implementations, demonstrating that generalized fractional operator theory can be systematically translated into practical controller synthesis. To the authors' knowledge, this work presents a unified framework based on nested weighted fractional operator compositions for the systematic construction of generalized Podlubny-type fractional PID controllers.

The main contributions of this paper can be summarized as follows:

- Establishes a unified composition-based framework for constructing nested weighted fractional operators.
- Introduces a novel pulling-type weighted fractional operator and rigorously analyzes its main analytical properties.
- Derives a generalized family of weighted Podlubny-type fractional PID controllers from the proposed framework.

The remainder of the paper is organized as follows. Section 2 reviews the fundamental concepts of fractional calculus and the composition-based viewpoint adopted throughout this work. Section 3 introduces the proposed nested weighted composition framework. Section 4 develops the generalized weighted fractional operators and establishes their main analytical properties. Section 5 presents the systematic synthesis of generalized Podlubny-type fractional PID controllers based on the proposed framework. Section 6 illustrates the proposed methodology through representative examples and discusses its implications for fractional-order controller design. Finally, Section 7 concludes the paper and outlines directions for future research.

2. Fractional calculus

In this section we recall the main definitions and properties of fractional calculus used throughout the paper. We adopt a uniform notation based on the Riemann–Liouville and Caputo operators.

2.1. Riemann–Liouville fractional integral

Definition 1 (***Riemann–Liouville Fractional Integral***)**.** Let $\lambda > 0$ and $f \in L^1(a,b)$. The Riemann–Liouville fractional integral of order λ is defined by

$$(I_t^{\lambda} f)(t) = \frac{1}{\Gamma(\lambda)} \int_a^t (t-\tau)^{\lambda-1} f(\tau)\, d\tau, \qquad t > a. \tag{1}$$

We denote by $I_t^{\lambda}(L^1)$ the class of functions representable as fractional integrals of summable functions, that is, $f = I_t^{\lambda}\varphi$, $\varphi \in L^1(a,b)$ (see [19–23]). The operator I_t^{α} is understood as the classical Riemann–Liouville fractional integral of order α. This convention is adopted to simplify notation and to emphasize the convolutional structure of the proposed framework.

2.2. Riemann–Liouville fractional derivative

Definition 2 (*Riemann–Liouville Fractional Derivative*). Let $\mu > 0$ and let $s = [\mu] + 1$. The Riemann–Liouville fractional derivative of order μ is defined by

$$(D_t^{\mu} f)(t) = \left(\frac{d}{dt}\right)^s (I_t^{s-\mu} f)(t), \qquad t > a. \tag{2}$$

For $0 < \mu < 1$, this reduces to

$$(D_t^{\mu} f)(t) = \frac{d}{dt}(I_t^{1-\mu} f)(t). \tag{3}$$

Remark 1. A function $f \in L^1(a, b)$ is said to have a summable fractional derivative if

$$I_t^{s-\mu} f \in AC^s([a, b]),$$

where $AC^s([a, b])$ denotes the space of functions whose $(s-1)$-th derivative is absolutely continuous on $[a, b]$ (see [19,20,24–26]).

The action of fractional operators on power functions provides explicit examples and serves as a fundamental computational tool [27,28].

Example 1. Let $\lambda, \mu \in (0, 1)$ and $\beta > -1$. Then

$$I_t^{\lambda}(t-a)^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta+\lambda+1)}(t-a)^{\beta+\lambda},$$

and

$$D_t^{\mu}(t-a)^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta-\mu+1)}(t-a)^{\beta-\mu},$$

provided that $\beta \neq \mu - 1$ (see [23,29]).

Remark 2. A notable feature of the Riemann–Liouville derivative is that constants are not annihilated. In particular,

$$D_t^{\mu} 1 = \frac{(t-a)^{-\mu}}{\Gamma(1-\mu)}.$$

This distinguishes the Riemann–Liouville framework from the classical derivative and from Caputo-type formulations. However, in the classical limit one recovers

$$\lim_{\mu \to 1} D_t^{\mu} 1 = 0,$$

which restores the standard annihilation property (see [19–21,23]).

2.3. Basic properties

In this subsection we collect some fundamental properties of the weighted fractional framework. Throughout this subsection, we assume:

- $f, g : [a, b] \to \mathbb{R}$ are sufficiently smooth functions;
- $f \in L^1(a, b)$ and the corresponding Riemann–Liouville fractional integrals and derivatives exist;
- whenever composition properties are considered, the required initial data are well-defined;
- for product rules, f and g are assumed analytic in a neighborhood of $[a, b]$.

These assumptions guarantee that all operators below are well-defined [19–23,30]).

Limit Case. Under the previous assumptions, the Riemann–Liouville derivative recovers the classical derivative in the integer-order limit

$$\lim_{\mu \to 1} D_t^{\mu} f(t) = \frac{d}{dt} f(t).$$

This property ensures consistency of the fractional framework with the classical differential setting (see [19,20,23,30])

Semigroup Property. In general, the Riemann–Liouville derivative does not satisfy the semigroup law

$$D_t^{\mu} D_t^{\beta} f \neq D_t^{\mu+\beta} f.$$

However, the semigroup property

$$D_t^{\mu} D_t^{\beta} f = D_t^{\mu+\beta} f \tag{4}$$

holds whenever $f^{(j)}(a) = 0, j = 0, \ldots, s-1$, with $s = \lceil \beta \rceil$, and f possesses sufficient regularity.

This condition will play an important role when constructing nested weighted operators.

2.4. Caputo fractional derivative

Besides the Riemann–Liouville formulation, the Caputo derivative plays a central role in control applications owing to its compatibility with classical initial conditions.

Definition 3 (***Caputo Derivative***)**.** Let $\mu > 0$ and $m = [\mu]$. The Caputo fractional derivative is defined by

$$ {}_cD_t^{\mu} f(t) = I_t^{m-\mu} \left(\frac{d}{dt} \right)^m f(t), \tag{5}$$

provided $f^{(m)} \in L^1(a, b)$.

For $0 < \mu < 1$, ${}_cD_t^{\mu} f(t) = I_t^{1-\mu} f'(t)$. Unlike the Riemann–Liouville derivative, ${}_cD_t^{\mu} 1 = 0$ (see [31]).

2.5. Laplace transform

For control applications, the Laplace transform provides the natural frequency-domain representation of fractional operators. Assuming $a = 0$ and sufficient regularity [22,30,32], the following relations hold

Fractional Integral:

$$ \mathcal{L}\{ I_t^{\lambda} f(t) \}(s) = s^{-\lambda} F(s). $$

Riemann–Liouville Derivative:

$$ \mathcal{L}\{ D_t^{\mu} f(t) \}(s) = s^{\mu} F(s) - \sum_{k=0}^{s-1} s^k \left[D_t^{\mu-k-1} f(t) \right]_{t=0^+}. \tag{6}$$

Under zero initial conditions, $\mathcal{L}\{ D_t^{\mu} f(t) \}(s) = s^{\mu} F(s)$.

Caputo Derivative (zero initial conditions):

$$ \mathcal{L}\{ {}_cD_t^{\mu} f(t) \}(s) = s^{\mu} F(s). $$

These classical operators constitute the elementary building blocks of the nested constructions introduced in the next section.

2.6. Convolution structure and fundamental properties

The convolution product provides the algebraic foundation in the construction of the weighted fractional operators introduced in this work. It provides the natural algebraic mechanism through which memory kernels interact and allows a unified representation of several generalized integral and differential operators (see [33–35]).

Definition 4 (***Convolution Product***)**.** Let $f, g \in L^1_{\text{loc}}([a, b])$. The convolution of f and g is defined by

$$ (f * g)(t) := \int_a^t f(t - \tau) g(\tau)\, d\tau, \qquad t > a. \tag{7}$$

Proposition 1 (***Differentiation of the Convolution***)**.** *Let f, g be sufficiently smooth functions such that the convolution $(f * g)(t)$ is well-defined. Then,*

$$ \frac{d}{dt}(f * g)(t) = f(0) g(t) + (f' * g)(t), \tag{8}$$

where f' denotes the classical derivative of f. Equivalently,

$$ \frac{d}{dt}(f * g)(t) = g(0) f(t) + (f * g')(t). \tag{9}$$

Corollary 1 (***Symmetric Differentiation Rule***)**.** *Let f, g be sufficiently smooth such that $(f * g)(t)$ is well-defined. Then the derivative of the convolution can be written in the symmetric form*

$$ \frac{d}{dt}(f * g)(t) = \frac{1}{2} \Big((f' * g)(t) + (f * g')(t) \Big). \tag{10}$$

Proof. From the classical differentiation formulas,

$$ \frac{d}{dt}(f * g)(t) = f(0) g(t) + (f' * g)(t), \tag{11}$$

and

$$ \frac{d}{dt}(f * g)(t) = g(0) f(t) + (f * g')(t). \tag{12}$$

Adding both expressions and dividing by two yields

$$\frac{d}{dt}(f * g)(t) = \frac{f(0)g(t) + g(0)f(t)}{2} + \frac{1}{2}\Big((f' * g)(t) + (f * g')(t)\Big). \tag{13}$$

In particular, if the boundary terms vanish, namely

$$f(0) = g(0) = 0,$$

the symmetric representation follows

$$\frac{d}{dt}(f * g)(t) = \frac{1}{2}\Big((f' * g)(t) + (f * g')(t)\Big). \tag{14}$$

These properties establish the convolution algebra that underlies the generalized weighted compositions developed in the next sections.

2.6.1. Convolution representation

The fractional integral admits the convolution form

$$I_t^{1-\mu} f = k_\mu * f, \qquad k_\mu(t) = \frac{t^{-\mu}}{\Gamma(1-\mu)}.$$

This representation will be essential when analyzing Laplace transforms of the weighted operators introduced later (see [22]).

2.7. Composition framework

Let Θ_d denote a linear differential operator and Θ_i a linear integral operator acting on a suitable function space.

Definition 5 (***Composition Operator***)**.** Let $k \in \mathbb{N}$. The composition operator of depth k is defined by

$$C_k = \Theta_{\sigma_1} \circ \Theta_{\sigma_2} \circ \cdots \circ \Theta_{\sigma_k}, \qquad \Theta_{\sigma_j} \in \{\Theta_d, \Theta_i\},$$

where $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_k)$ is an arbitrary sequence o differential and integral operators.

The above definition allows the construction of a broad family of operators generated from differential and integral building blocks. Typical examples include

$$\begin{aligned} &\Theta_d \circ \Theta_i \circ \Theta_d, \\ &\Theta_i \circ \Theta_d \circ \Theta_i, \\ &\Theta_d \circ \Theta_d \circ \Theta_i \circ \Theta_i, \end{aligned} \tag{15}$$

as well as repeated structures such as $(\Theta_d \circ \Theta_i)^k$, $(\Theta_i \circ \Theta_d)^k$. The ordering of differentiation and integration determines the resulting memory structure of the operator. Consequently, different composition chains may generate substantially different dynamical behaviors.

Remark 3. The present work focuses on the construction and application of nested weighted fractional operators. A detailed study of algebraic properties such as commutativity, associativity, and the equivalence of different composition chains is left for future research. Such an analysis could provide a more complete characterization of the operator algebra generated by the proposed framework.

2.8. Recovery of classical fractional operators

The compositional viewpoint naturally recovers several classical fractional operators as particular arrangements of differential and integral operators.

Definition 6. Let $\mu \in (0,1)$ and let $n, m \in \{0,1\}$. Let $f \in L^1(a,b)$ be such that $\left(\frac{d}{dt}\right)^m f \in L^1(a,b)$, and $I_a^{1-\mu}\left(\frac{d}{dt}\right)^m f \in AC([a,b])$. We define

$$C^{\mu}_{n,m} f(t) := \left(\frac{d}{dt}\right)^n \circ I_a^{1-\mu} \circ \left(\frac{d}{dt}\right)^m f(t), \qquad t > a.$$

Recovery of classical operators

The parameters n and m determine the position of the integer-order differentiation relative to the fractional integral. Different choices recover several well-known operators:

- If $(n, m) = (1, 0)$,

$$C^{\mu}_{1,0} f = \frac{d}{dt}\left(I^{1-\mu}_{a} f\right) = D^{\mu}_{a} f,$$

which corresponds to the Riemann–Liouville fractional derivative.

- If $(n, m) = (0, 1)$,

$$C^{\mu}_{0,1} f = I^{1-\mu}_{a}\left(\frac{d}{dt} f\right) = {}^{C}D^{\mu}_{a} f,$$

which corresponds to the Caputo fractional derivative.

- If $(n, m) = (0, 0)$,

$$C^{\mu}_{0,0} f = I^{1-\mu}_{a} f,$$

which corresponds to a fractional integral operator.

Therefore, classical fractional operators arise naturally as particular configurations of a differential–integral composition chain.

Remark 4. The compositional structure introduced above is not restricted to discrete configurations determined by $(n, m) \in \{0, 1\}$. More general operators can be obtained by redistributing fractional integrals around the differentiation stage.

Hilfer operator as a structured composition

Let $\mu \in (0, 1)$ and $\nu \in [0, 1]$. Assume that all expressions below are well-defined. We define (see [36])

$$C^{\mu}_{\nu} f(t) := I^{\nu(1-\mu)}_{a} \circ \frac{d}{dt} \circ I^{(1-\nu)(1-\mu)}_{a} f(t), \qquad t > a.$$

Remark 5. The operator C^{μ}_{ν} coincides with the Hilfer fractional derivative and provides a continuous transition between the Riemann–Liouville and Caputo definitions. Indeed, $\nu = 0$. Then $C^{\mu}_{0} = D^{\mu}_{a}$, while $\nu = 1$. Then $C^{\mu}_{1} = {}^{C}D^{\mu}_{a}$. Consequently, the Hilfer derivative may be interpreted as a structured redistribution of fractional integration stages around the differentiation operator.

Katugampola operator as a weighted composition

The compositional philosophy can also be extended to generalized fractional operators involving deformation parameters and weighted differential structures. A representative example is the Katugampola derivative, which can be written in the structural form (see [37])

$$D^{\mu,\rho}_{a} f(t) = t^{1-\rho} \frac{d}{dt}\left(I^{1-\mu,\rho}_{a} f(t)\right), \qquad \rho > 0,$$

where $I^{1-\mu,\rho}_{a}$ denotes the generalized Katugampola fractional integral. The above expression reveals that the Katugampola derivative can be interpreted as a weighted differential operator acting on a generalized fractional integral. Therefore, it remains compatible with the composition-based framework developed in this section.

The previous examples show that several classical and generalized fractional operators can be interpreted as structured compositions of differential and integral operators. In particular, Riemann–Liouville and Caputo operators arise from simple compositions, Hilfer operators emerge through a redistribution of fractional integration stages, and Katugampola-type operators introduce weighted deformations while preserving the underlying compositional structure. This observation suggests that operator composition plays a more fundamental role than any individual fractional definition. Motivated by this point of view, the next section introduces weighted composition chains, where each stage of the operator may be modulated through suitable weighting functions.

3. Nested weighted composition chains

The Katugampola derivative illustrates that weighted differential structures can be naturally incorporated into a composition framework. Motivated by this observation, we consider a broader class of weighted composition chains in which both the differential and integral stages are modulated through weighting functions.

In contrast with classical fractional operators, the weighting functions may depend not only on time and the fractional order, but also on auxiliary signals associated with the dynamical system under consideration. This additional flexibility will allow for the construction of fractional PID-type operators and control-oriented architectures later. To this end, we introduce the weighted differential block

$$\Theta_{d} := \left(q_1(t, \overline{\mu}, p)\frac{d}{dt} - \gamma_1(t, \overline{\mu}, p)\right)^{n_1},$$

where $n_1 \in \mathbb{N}$, q_1 and γ_1 are sufficiently smooth functions, $\overline{\mu} = \{\mu_i\}_{i=1}^{N}$ denotes a collection of fractional orders to represent and p denotes an auxiliary function. Similarly, we define the weighted integral block

$$\Theta_{i} := \left(q_2(t, \overline{\mu}, g)\sum_{i=1}^{N} \theta_i {I_t}^{1-\mu_i} - \gamma_2(t, \overline{\mu}, g)\right)^{n_2},$$

where $\theta_i \in \mathbb{R}$, $n_2 \in \mathbb{N}$, q_2 and γ_2 are sufficiently smooth functions, g is an auxiliary function. Therefore, the general weighted composition chain takes the form

$$\mathcal{NW}_{\bar{n}}^{\bar{\mu}} = \left(q_{11}(t,\bar{\mu},p)\frac{d}{dt} - \gamma_{11}(t,\bar{\mu},p)\right)^{n_{11}} \circ \left(q_{21}(t,\bar{\mu},g)\sum_{i=1}^{N}\theta_{i1}I_t^{1-\mu_i} - \gamma_{21}(t,\bar{\mu},g)\right)^{n_{21}} \circ \cdots \cdots \circ \left(q_{1j}(t,\bar{\mu},p)\frac{d}{dt} - \gamma_{1j}(t,\bar{\mu},p)\right)^{n_{1j}} \circ \left(q_{2j}(t,\bar{\mu},g)\sum_{i=1}^{N}\theta_{ij}I_t^{1-\mu_i} - \gamma_{2j}(t,\bar{\mu},g)\right)^{n_{2j}}. \tag{16}$$

To make explicit the notation used in the nested composition, let $\bar{\mu} = (\mu_1, \mu_2, \ldots, \mu_N)$, $0 < \mu_i < 1$, denote the vector of fractional-order parameters, where N denotes the number of fractional-memory channels, and let $\bar{n} = (n_{11}, n_{21}, \ldots, n_{1J}, n_{2J}) \in \mathbb{N}^{2J}$ denote the collection of integer-valued exponents associated with the differential and integral blocks at each composition level, where J denotes the number of nested composition levels. For each $j = 1, \ldots, J$, the functions q_{1j} and q_{2j} are real-valued weighting functions, whereas γ_{1j} and γ_{2j} are real-valued shift functions, with the regularity required for the corresponding differential or fractional integral operations to be well-defined. Likewise, $\theta_{ij} \in \mathbb{R}, \quad i = 1, \ldots, N, \quad j = 1, \ldots, J$, are real-valued coefficients that determine the contribution of each fractional-memory channel at the jth composition level. The regularity assumptions on the functions q_{1j}, q_{2j}, γ_{1j}, and γ_{2j} are understood to be imposed according to the analytical property under consideration.

This generalized chain provides a hierarchical mechanism for combining differential and memory effects across multiple nested levels. Each stage of composition j introduces a local interaction between weighted differentiation and weighted fractional memory, while repeated nesting allows these effects to propagate through successive layers of operators.

The finite family of fractional orders $\{\mu_i\}_{i=1}^{N}$ defines the memory spectrum of the operator. In particular, the parameter N determines the number of simultaneous memory integrals, allowing the coexistence of short-term and long-term memory effects. The coefficients θ_{ij} modulate the contribution of each memory integral at every composition level, producing a nonuniform distribution of memory influence across the chain.

On the other hand, the integer exponents n_{1j}, n_{2j}, determine the local structural depth of the differential and integral blocks. Larger values of n_{1j} increase the local differential order, whereas larger values of n_{2j} amplify the internal memory accumulation. Consequently, the interplay between the composition depth j, the number of memory channels N, and the exponents n_{1j}, n_{2j} generates a highly flexible operator family capable of reproducing a broad range of dynamical behaviors.
The previous discussion leads to the following structural interpretation of the generalized composition chain:

- For $N = 1$ and $n_{1j} = n_{2j} = 1$, classical weighted fractional operators are recovered;
- Increasing N enriches the memory structure by introducing multiple fractional layers;
- Increasing j deepens the nested composition hierarchy, producing multistage memory-differential interactions;
- Increasing n_{1j} or n_{2j} intensifies the local action of differentiation or memory accumulation.

In control applications, the auxiliary functions p and g will typically be associated with tracking errors (see [38]), state variables, or other closed-loop signals. This dependence allows the weighting functions and shifts to adapt dynamically to the system evolution, leading to nonlinear and time-varying fractional control architectures. At the present stage, however, these functions are kept general in order to preserve the operator-theoretic nature of the construction.

Example 2 (***Nested Weighted generalized Riemann Liouville Structure***).
Choosing $n_1 = 1$, $n_2 = 1$, $\gamma_1 = \gamma_2 = 0$, gives

$$\Theta_d \circ \Theta_i = q_1(t,\bar{\mu},p)\frac{d}{dt}\left(q_2(t,\bar{\mu},g)\sum_{i=1}^{N} I_t^{1-\mu_i}\right).$$

For $q_1 = q_2 = 1, N = 1$, the classical Riemann–Liouville derivative is recovered.

Example 3. Under the appropriate hypotheses, consider

$$\Theta_d \circ \Theta_i f = \frac{d}{dt} \circ \left(t^{\rho} I_t^{1-\mu}\right) f = t^{\rho} D_a^{\mu} f + \rho t^{\rho-1} I_t^{1-\mu} f., \qquad \rho \in \mathbb{R}.$$

This operator differs from classical Riemann–Liouville and Caputo derivatives and represents a time-varying fractional dynamics, which is particularly relevant in control applications with variable gains.

This construction is related to generalized weighted fractional operators appearing in the literature, including the Katugampola derivative, fractional derivatives with respect to another function, and conjugation-based fractional operators. However, the present operator arises naturally from the composition-chain framework developed here, providing a direct structural interpretation of variable-coefficient fractional dynamics [37,39].

Example 4 (***The triple nested configuration***). Let $q_i(t,\bar{\mu},f_i) = q_i$ and $\gamma_i(t,\bar{\mu},g_i) = \gamma_i$ be sufficiently smooth functions, $n_1, n_2, n_3 \in \mathbb{N}$. We define the nested weighted operator

$$\Theta_d \circ \Theta_i \circ \Theta_d = \left(q_1\frac{d}{dt} - \gamma_1\right)^{n_1}\left(q_2 I_a^{1-\mu} - \gamma_2\right)^{n_2}\left(q_3\frac{d}{dt} - \gamma_3\right)^{n_3}.$$

4. Weighted fractional operators

The weighted structure of the Katugampola derivative reveals that fractional operators may be enriched by introducing nontrivial weighting mechanisms into their differential–integral composition. Motivated by this observation, and seeking greater structural richness and flexibility, particularly for the construction of PID-type dynamical models, we consider a broader class of generalized weighted fractional operators.

In contrast to classical fractional definitions, the proposed formulation incorporates multiple memory layers, variable weighting functions, and auxiliary functions (see [40,41]). This allows the simultaneous interaction of several memory integrals, each with potentially different fractional orders, while the weighting functions provide an additional mechanism for modulating their influence. Such a framework naturally extends the weighted philosophy introduced by Almeida [39,42], Contreras [43], Fernandez [16], and Katugampola [37,44], while considerably enlarging its structural capacity. In particular, this richer architecture provides a suitable foundation for generating fractional PID-type operators and adaptive multi-memory control laws.

Definition 7 (*Weighted Fractional Operator*).
Let $\overline{\mu} = (\mu_1, \mu_2, \dots, \mu_N), 0 < \mu_i < 1$, and let $\theta_i \in \mathbb{R}, i = 1, \dots, N$. Assume that $q_1 = q_1(t, \overline{\mu}, p(t)), q_2 = q_2(t, \overline{\mu}, g(t))$, are sufficiently smooth weighting functions, where p and g are auxiliary functions. The weighted fractional operator is defined as

$$\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = \mathcal{N}\mathcal{W}_{\overline{n}}^{\overline{\mu}} f(t) = q_1(t, \overline{\mu}, p(t)) \frac{d}{dt} \left(q_2(t, \overline{\mu}, g(t)) \sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} f(t) \right).$$

Remark 6 (*Auxiliary Functions*).
The functions p and g play the role of auxiliary variables that may be selected according to the application under consideration. For the theoretical developments presented in this section they are assumed sufficiently smooth so that all derivatives and fractional integrals are well-defined. In control applications, these auxiliary variables will later be associated with the tracking error and other closed-loop signals, allowing the weighting functions to adapt the operator structure to the system dynamics.

Remark 7 (*Motivational Structure*).
Although the general compositional framework introduced in the previous section allows arbitrarily deep nested configurations, in this work we restrict our analysis to the weighted operator defined above. This choice is motivated by two main reasons. First, the proposed structure already contains enough flexibility to generate a broad family of fractional proportional–integral–derivative (PID) models through suitable choices of the coefficients β_i, the fractional orders μ_i, and the weighting functions q_1 and q_2. Second, this simplified configuration preserves the essential interplay between differentiation and memory while keeping the resulting operators tractable for analysis, identification, and control implementation.

In this sense, the weighting functions provide an additional degree of freedom that extends classical fractional PID constructions, allowing adaptive, nonlinear, and multi-memory effects without increasing the structural complexity of the operator chain.

4.1. Basic properties

- Linearity: Assume that the weighting functions $q_1 = q_1(t, \overline{\mu})$, $q_2 = q_2(t, \overline{\mu})$ are prescribed and independent of the state. Then the operator ${}_qD_a^{\overline{\mu}}$ is linear. Let $c_1, c_2 \in \mathbb{R}$ and let u, v belong to the domain of the operator. Then

$$\mathcal{W}_t^{\boldsymbol{\mu}}(c_1 u + c_2 v) = q_1 \frac{d}{dt} \left[q_2 \sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} (c_1 u + c_2 v) \right].$$

  Using the linearity of the fractional integral, $I_t^{1-\mu_i}(c_1 u + c_2 v) = c_1 I_t^{1-\mu_i} u + c_2 I_t^{1-\mu_i} v$, we obtain

$$\mathcal{W}_t^{\boldsymbol{\mu}}(c_1 u + c_2 v) = c_1 q_1 \frac{d}{dt} \left[q_2 \sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} u \right] + c_2 q_1 \frac{d}{dt} \left[q_2 \sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} v \right].$$

  Therefore, $\mathcal{W}_t^{\boldsymbol{\mu}}(c_1 u + c_2 v) = c_1 \mathcal{W}_t^{\boldsymbol{\mu}} u + c_2 \mathcal{W}_t^{\boldsymbol{\mu}} v$.

Remark 8 (*Loss of Linearity*).
If $q_1 = q_1(t, \overline{\mu}, p(t))$, $q_2 = q_2(t, \overline{\mu}, g(t))$ depend explicitly on the state or auxiliary variables, the operator is not necessarily linear. Indeed, $q_1(t, \overline{\mu}, u + v) \neq q_1(t, \overline{\mu}, u) + q_1(t, \overline{\mu}, v)$ in general. Consequently, the proposed framework includes both linear and nonlinear weighted fractional operators as particular cases.

- Constant-Annihilating: Suppose that $N = 1$, $\beta_1 = 1$, and $q_2(t, \mu) = (t - a)^{\mu - 1}$. Then $\mathcal{W}_t^{\boldsymbol{\mu}} 1 = 0$. Using $I_t^{1-\mu} 1 = \dfrac{(t-a)^{1-\mu}}{\Gamma(2-\mu)}$, we obtain $q_2(t, \mu) I_t^{1-\mu} 1 = \dfrac{1}{\Gamma(2-\mu)}$. Since the above expression is constant, $\dfrac{d}{dt}\left(q_2(t, \mu) I_t^{1-\mu} 1 \right) = 0$. Hence, $\mathcal{W}_t^{\boldsymbol{\mu}} 1 = 0$.

- Constant-Annihilating Multi-integral Property: Suppose that for each $i = 1, \dots, N$, $q_{2,i}(t, \mu_i) = (t-a)^{\mu_i - 1}$. Then $\mathcal{W}_t^{\boldsymbol{\mu}} 1 = 0$. Indeed, using $I_t^{1-\mu_i} 1 = \frac{(t-a)^{1-\mu_i}}{\Gamma(2-\mu_i)}$, we obtain $q_{2,i}(t, \mu_i) I_t^{1-\mu_i} 1 = \dfrac{1}{\Gamma(2-\mu_i)}$, which is constant. Therefore, $\dfrac{d}{dt}\left(\sum_{i=1}^{N} \beta_i q_{2,i}(t, \mu_i) I_t^{1-\mu_i} 1\right) = 0$, and consequently, $\mathcal{W}_t^{\boldsymbol{\mu}} 1 = 0$.

Remark 9. The constant-annihilating property is not guaranteed for arbitrary weighting functions. Instead, it can be enforced through an appropriate selection of the weighting functions, as illustrated above. When this property is satisfied, a constant error contribution is annihilated by the corresponding fractional derivative channel, thereby avoiding a persistent contribution from that channel at steady state. Conversely, for a general Riemann–Liouville-based weighted operator without the constant-annihilating condition, a constant error need not be annihilated. Consequently, zero steady-state error cannot be inferred solely from the use of the Riemann–Liouville operator and must instead be established from the specific controller structure and the selected weighting functions.

- **Convolution Structure and Transfer Function Representation.** Let $q_1, q_2, f \in L^1_{\mathrm{loc}}(0, \infty)$ and consider the weighted fractional operator $\mathcal{W}_t^{\boldsymbol{\mu}}$ defined in Definition 7. The operator is given by

$$\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = q_1(t, \overline{\mu}, p(t)) \frac{d}{dt}\left(q_2(t, \overline{\mu}, g(t)) M_f(t)\right),$$

where

$$M_f(t) = \sum_{i=1}^{N} \theta_i \left(I_t^{1-\mu_i} f\right)(t),$$

with

$$0 < \mu_i < 1, \qquad \theta_i \in \mathbb{R}.$$

Using the Riemann–Liouville fractional integral representation, the memory term admits the convolution representation

$$M_f(t) = \sum_{i=1}^{N} \theta_i \left(k_{\mu_i} * f\right)(t),$$

where

$$k_{\mu_i}(t) = \frac{t^{-\mu_i}}{\Gamma(1-\mu_i)}.$$

The following statements hold.

(i) For constant weighting functions q_1 and q_2, the operator reduces to

$$\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = q_1 q_2 \frac{d}{dt} M_f(t).$$

Under zero initial conditions, its Laplace transform is $\mathcal{L}\left\{\mathcal{W}_t^{\boldsymbol{\mu}} f\right\}(s) = q_1 q_2\, s\, M_f(s)$. Since

$$M_f(s) = \left(\sum_{i=1}^{N} \theta_i s^{-(1-\mu_i)}\right) F(s),$$

we obtain

$$\mathcal{L}\left\{\mathcal{W}_t^{\boldsymbol{\mu}} f\right\}(s) = q_1 q_2 \left(\sum_{i=1}^{N} \theta_i s^{\mu_i}\right) F(s).$$

(ii) If at least one of the weighting functions depends explicitly on time,

$$\frac{d}{dt} q_1(t) \neq 0 \qquad \text{or} \qquad \frac{d}{dt} q_2(t) \neq 0,$$

then, in general, the operator is time-varying and does not admit a pure convolution representation. In particular, writing

$$\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = q_1(t) \frac{d}{dt}\left[q_2(t) M_f(t)\right],$$

the product rule gives

$$\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = q_1(t) q_2'(t) M_f(t) + q_1(t) q_2(t) M_f'(t).$$

Consequently, in general, there does not exist a multiplicative symbol $G(s)$ such that

$$\mathcal{L}\left\{\mathcal{W}_t^{\boldsymbol{\mu}} f\right\}(s) = G(s) F(s).$$

(iii) For constant weighting functions, the transfer-function representation follows from part (i). Defining

$$G_{\boldsymbol{\mu}}(s) = q_1 q_2 \sum_{i=1}^{N} \theta_i s^{\mu_i},$$

we have

$$\mathcal{L}\left\{\mathcal{W}_t^{\boldsymbol{\mu}} f\right\}(s) = G_{\boldsymbol{\mu}}(s) F(s). \tag{17}$$

Proof. Using the representation of the Riemann–Liouville fractional integral, for $0 < \mu_i < 1$ we have

$$\left(I_t^{1-\mu_i} f\right)(t) = \frac{1}{\Gamma(1-\mu_i)} \int_0^t (t-\tau)^{-\mu_i} f(\tau)\, d\tau.$$

Define

$$k_{\mu_i}(t) = \frac{t^{-\mu_i}}{\Gamma(1-\mu_i)}.$$

Hence, $\left(I_t^{1-\mu_i} f\right)(t) = \left(k_{\mu_i} * f\right)(t)$, and therefore

$$M_f(t) = \sum_{i=1}^{N} \theta_i \left(k_{\mu_i} * f\right)(t),$$

which establishes the convolution representation of the memory term.

For part (i), when q_1 and q_2 are constant, Definition 7 gives

$$\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = q_1 q_2 \frac{d}{dt} M_f(t).$$

Under zero initial conditions, the Laplace transform of the derivative is

$$\mathcal{L}\left\{\frac{d}{dt} M_f\right\}(s) = s M_f(s).$$

Moreover, using the Laplace transform of the Riemann–Liouville fractional integral,

$$M_f(s) = \sum_{i=1}^{N} \theta_i s^{-(1-\mu_i)} F(s) = \left(\sum_{i=1}^{N} \theta_i s^{-(1-\mu_i)}\right) F(s).$$

Consequently,

$$\begin{aligned}
\mathcal{L}\left\{\mathcal{W}_t^{\boldsymbol{\mu}} f\right\}(s) &= q_1 q_2 s M_f(s) \\
&= q_1 q_2 s \left(\sum_{i=1}^{N} \theta_i s^{-(1-\mu_i)}\right) F(s) \\
&= q_1 q_2 \left(\sum_{i=1}^{N} \theta_i s^{\mu_i}\right) F(s),
\end{aligned}$$

which proves part (i).

For part (ii), suppose that at least one of the weighting functions depends explicitly on time. Then

$$\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = q_1(t) \frac{d}{dt}\left[q_2(t) M_f(t)\right].$$

Applying the product rule yields $\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = q_1(t) q_2'(t) M_f(t) + q_1(t) q_2(t) M_f'(t)$. Thus, the operator contains products of time-dependent functions with the memory term and its derivative. In general, these products cannot be represented as a convolution with a kernel depending only on the time difference. Hence, the operator is generally time-varying and does not admit a multiplicative Laplace-domain representation of the form $\mathcal{L}\left\{\mathcal{W}_t^{\boldsymbol{\mu}} f\right\}(s) = G(s) F(s)$. This proves part (ii).

Finally, for constant weighting functions, part (i) gives

$$\mathcal{L}\left\{\mathcal{W}_t^{\boldsymbol{\mu}} f\right\}(s) = q_1 q_2 \left(\sum_{i=1}^{N} \theta_i s^{\mu_i}\right) F(s).$$

Therefore, defining $G_{\boldsymbol{\mu}}(s) = q_1 q_2 \sum_{i=1}^{N} \theta_i s^{\mu_i}$, we obtain

$$\mathcal{L}\left\{\mathcal{W}_t^{\boldsymbol{\mu}} f\right\}(s) = G_{\boldsymbol{\mu}}(s) F(s),$$

which proves part (iii).

Remark 10. The convolution representation above refers specifically to the fractional-memory term $M_f(t)$ through the Riemann–Liouville fractional–integral kernels k_{μ_i}. The weighting functions q_1 and q_2 in Definition 7 act multiplicatively on the memory term and are not themselves interpreted as convolution kernels. Accordingly, the multiplicative transfer-function representation is restricted to the case of time-independent weighting functions.

4.2. Illustrative examples

Example 5 (*Multi-Memory PID Structure*). Let $N = 3$, let $f \in L^1(a,b)$, $q_2 \in C^1(a,b)$, $\theta_1, \theta_2, \theta_3 \in \mathbb{R}$ and choose $\mu_1 = \alpha, \mu_2 = \mu, \mu_3 = \nu, 0 < \alpha, \mu, \nu < 1$. Then

$$\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = q_1(t) \frac{d}{dt} \left[q_2(t) \left(\theta_1 I_t^{1-\alpha} f(t) + \theta_2 I_t^{1-\mu} f(t) + \theta_3 I_t^{1-\nu} f(t) \right) \right].$$

This structure contains three independent fractional-memory integrals and serves as a natural building block for generalized PID-type architectures.

Example 6 (*Polynomial Weights in Convolutional Realization*). Choose $q_1(t) = (1+t)^r, q_2(t) = (1+t)^s$ for $t > 0$, $r, s \in \mathbb{R}$, $\theta_i \in \mathbb{R}$, and let $f \in L^1_{\mathrm{loc}}(0,\infty)$. Assume that the convolution $q_2 * \left(\sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} f \right)$ is locally absolutely continuous on $(0,\infty)$. Then, the convolutional realization is given by

$$\mathcal{W}_{t,\mathrm{conv}}^{\boldsymbol{\mu}} f(t) = (1+t)^r * \frac{d}{dt} \left[(1+t)^s * \sum_{i=1}^{N} \theta_i \left(I_t^{1-\mu_i} f \right)(t) \right] = (1+t)^r * \left[(1+t)^s * \sum_{i=1}^{N} \theta_i \left(D_t^{\mu_i} f \right)(t) \right].$$

The parameters r and s control the growth or attenuation of the weighting kernels and are particularly convenient for visualization purposes.

Example 7 (*Exponential Weights depending on $\overline{\mu}$ in Convolutional Realization*).
Let $N = 2$ Choose $q_1(t) = \mu_1 e^{-at}$, $q_2(t) = e^{-b\mu_2 t}$, with $a, b > 0$ and $\theta_1, \theta_2 \in \mathbb{R}$, for $f \in L^1(a,b)$. Then

$$\mathcal{W}_{t,\mathrm{conv}}^{\boldsymbol{\mu}} f = \mu_1 e^{-at} * \frac{d}{dt} \left[e^{-b\mu_2 t} * \left(\theta_1 I_t^{1-\alpha} + \theta_2 I_t^{1-\mu} \right) f \right].$$

This choice introduces an exponentially decaying memory contribution and will be used later for numerical illustrations and control implementations.

Example 8 (*Weighted Caputo in Convolution Form*). Let $f \in C^1(a,b)$. Consider the weighted Caputo-type operator

$${}^c\mathcal{W}_t^{\boldsymbol{\mu}} f(t) = I_t^{1-\mu} \left(q(t,\mu) \frac{d}{dt} f(t) \right).$$

Using the convolution representation of the Riemann–Liouville fractional integral, it can be written as

$${}^c\mathcal{W}_t^{\boldsymbol{\mu}} f = k_\mu * \left(q f' \right).$$

If q is constant, then

$${}^c\mathcal{W}_t^{\boldsymbol{\mu}} f = q \left(k_\mu * f' \right),$$

and under zero initial conditions,

$$\mathcal{L} \left\{ {}^c\mathcal{W}_t^{\boldsymbol{\mu}} f \right\}(s) = q s^\mu F(s).$$

Remark 11. If the weight function $q(t,\mu)$ depends explicitly on time, then in general

$$k_\mu * (q f') \neq (q k_\mu) * f',$$

and the operator is no longer a pure convolution operator. In this case, the Laplace transform does not reduce to a simple multiplicative symbol $q(s)s^\mu$, but instead involves a convolution in the Laplace domain or a pseudo-differential symbol depending on the joint structure of q and the kernel k_μ. Therefore, the representation

$$\mathcal{L}\{ {}^c\mathcal{W}_t^{\boldsymbol{\mu}} f \}(s) = q(s) s^\mu F(s)$$

is rigorously valid when the weight is constant (or sufficiently slowly varying so that convolution structure is preserved) and zero initial conditions are assumed.

Example 9 (*Numerical Implementation via Grünwald–Letnikov Approximation*).
To illustrate the numerical behavior of the proposed weighted fractional operator, we consider an implementation based on the Grünwald–Letnikov approximation (see [22,30]) combined with discrete convolution. Consider the input function $f(t) = \sin(2t), t \in [0,10]$, with the weighting functions $q_1(t) = e^{-0.3t}, q_2(t) = \cos t$.

Let $N = 3$, with fractional orders $\mu_1 = 0.2, \mu_2 = 0.5, \mu_3 = 0.8$, and coefficients $\theta_1 = 1, \theta_2 = -0.5, \theta_3 = 0.8$. For comparison purposes, we consider two realizations of the weighted fractional structure. The convolutional realization is defined as

$$\mathcal{W}_{t,\mathrm{conv}}^{\boldsymbol{\mu}} f(t) = q_1(t) * \frac{d}{dt} \left[q_2(t) * \sum_{i=1}^{3} \theta_i \left(I_t^{1-\mu_i} f \right)(t) \right],$$

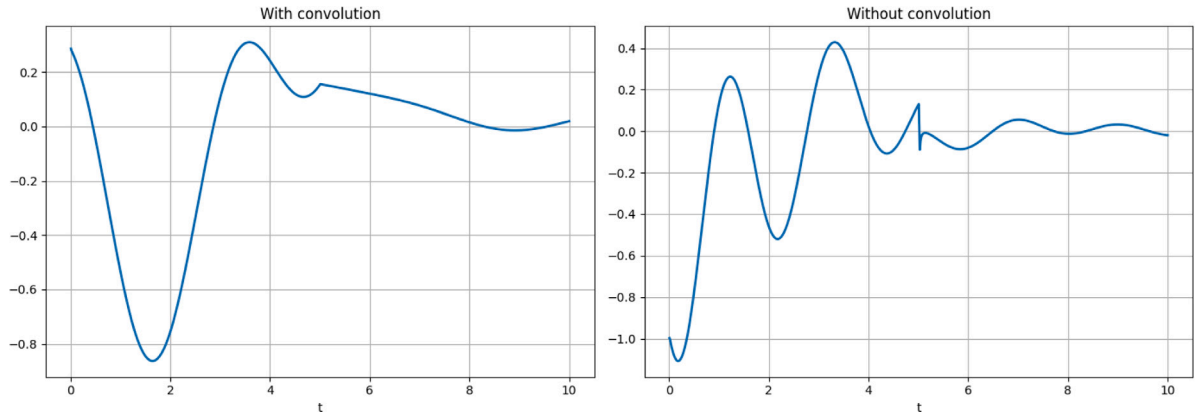


Fig. 1. Numerical comparison of the two realizations considered in Example 9 for $f(t) = \sin(2t)$. The left panel shows the convolutional realization, while the right panel shows the multiplicative realization. Both realizations use the same fractional orders, coefficients, and weighting functions.

whereas the multiplicative realization, consistent with Definition 7, is given by

$$\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{mult}} f(t) = q_1(t) \frac{d}{dt} \left[q_2(t) \sum_{i=1}^{3} \theta_i \left(I_t^{1-\mu_i} f \right)(t) \right].$$

Both realizations use the same input function, fractional orders, coefficients, and weighting functions. The numerical simulations allow us to compare the effect of treating the weighting functions as convolution kernels with their multiplicative use in the weighted fractional operator.

Fig. 1 illustrates the numerical results obtained for the two realizations of the weighted fractional operator. As shown in the figure, the convolutional realization introduces additional smoothing and delayed memory effects compared with the multiplicative realization. This behavior is associated with the convolution kernels distributing the corresponding contributions over the history of the signal.

5. PID-type operators generated by weighted fractional compositions

The objective of this section is to demonstrate that the weighted fractional operators introduced previously naturally generate a broad class of PID-type control structures. By appropriately selecting the memory orders, weights, and convolution kernels, the proposed framework recovers classical PID controllers, fractional Podlubny-type controllers, and new multi-memory architectures (see [1,22,45]).

Throughout this section we assume: $f \in AC^1([0,T])$, $\mu_i \in (0,1)$, $\beta_i, \lambda \in \mathbb{R}$, $q_1, q_2 \in L^1(0,T)$, all convolutions are well-defined. The adaptive weighting functions are assumed to have the general form

$$q_1 = q_1(t, e, \overline{\mu}), \qquad q_2 = q_2(t, e, \overline{\mu}),$$

where $e(t)$ denotes the tracking error and $\overline{\mu}$ represents the fractional tuning parameters of the control architecture (see [38,46,47]).

5.1. General fractional PID generator

We begin by introducing the fundamental operator from which all subsequent PID-type structures will be generated.

Definition 8 (***Nested Weighted Fractional Operator***).
Let $\overline{\mu} = (\mu_1, \mu_2, \ldots, \mu_N)$, $0 < \mu_i < 1$, and let $\beta_i, \lambda \in \mathbb{R}$, $i = 1, \ldots, N$. Assume that $q_1, q_2 \in AC([0,T]) \cap L^1(0,T)$, and let $f \in AC([0,T]) \cap L^1(0,T)$ be such that all the fractional integrals and derivatives exist. We define the nested weighted fractional operator by

$$\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f(t) = q_1(t) * \frac{d}{dt} \left(q_2(t) * \sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} \right) f(t) + \lambda f(t), \tag{18}$$

where $*$ denotes the convolution product.

The operator $\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}}$ combines proportional, integral, and fractional-memory effects within a single weighted convolutional framework. Different controller architectures can be generated through suitable choices of the coefficients β_i, the memory orders μ_i, and the proportional parameter λ.

Using the symmetric convolutional differentiation identity $(f * g)' = \frac{1}{2}\left(f' * g + f * g'\right)$, which follows directly from the standard differentiation property of convolutions, the operator (18) can be rewritten as

$$\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f(t) = \frac{1}{2} q_1(t) * \left(\frac{dq_2(t)}{dt} * \sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} f(t) + q_2(t) * \sum_{i=1}^{N} \theta_i D_t^{\mu_i} f(t) \right) + \lambda f(t). \tag{19}$$

This representation provides a Leibniz-type decomposition of the operator: the first term captures the interaction between the variation of the weighting kernel and the memory contribution, while the second term acts directly on the fractional derivative structure.

Applying the Laplace transform to (19), and using the standard convolution and fractional integral properties, we obtain

$$\mathcal{L}\left\{ \mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f \right\}(s) = \frac{1}{2} Q_1(s) \left(s Q_2(s) \sum_{i=1}^{N} \theta_i s^{-(1-\mu_i)} + Q_2(s) \sum_{i=1}^{N} \theta_i s^{\mu_i} \right) F(s) + \lambda F(s). \tag{20}$$

Simplifying,

$$\mathcal{L}\left\{ \mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f \right\}(s) = \left(\frac{Q_1(s) Q_2(s)}{2} \sum_{i=1}^{N} \theta_i s^{\mu_i} + \lambda \right) F(s). \tag{21}$$

This representation clearly shows how the weights influence the multi-order fractional dynamics of the operator.

5.1.1. PI and PD-type structures generated by the nested weighted operator

Starting from the general operator (18), different PID-type architectures arise naturally through suitable choices of the memory orders, weighting functions, and coefficients. For frequency-domain analysis, we consider weighting functions of the form $q_1 = q_1(t, \overline{\mu}), q_2 = q_2(t, \overline{\mu})$, and assume that, for fixed q_1 and q_2 are locally integrable and possess Laplace transforms in a common right half-plane, $\overline{\mu} \in \mathbb{Z}$. Under these assumptions, we denote their Laplace transforms by $Q_1(s, \overline{\mu}) = \mathcal{L}\{q_1(\cdot, \overline{\mu})\}(s)$, $Q_2(s, \overline{\mu}) = \mathcal{L}\{q_2(\cdot, \overline{\mu})\}(s)$. The resulting expressions provide a general Laplace-domain representation of the weighted fractional operator, without requiring a specific functional form for the weighting functions.

The multiplicative transfer-function representation is obtained only under the additional conditions that the weighting factors enter the operator through time-invariant convolutional or multiplicative structures for which the corresponding Laplace-domain factorization is valid. For general time-dependent weighting functions, the Laplace transform of the products does not, in general, factorize into $Q_1(s)Q_2(s)$, and the resulting operator must instead be treated as a time-varying weighted operator.

5.1.2. Integral representation

The original operator (18) naturally provides a pure integral-memory representation

$$\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f(t) = q_1(t, \overline{\mu}) * \frac{d}{dt} \left(q_2(t, \overline{\mu}) * \sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} f(t) \right) + \lambda f(t). \tag{22}$$

Using (17), its Laplace transform is given by

$$\mathcal{L}\left\{ \mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f \right\}(s) = \left(Q_1(s, \overline{\mu}) Q_2(s, \overline{\mu}) \sum_{i=1}^{N} \theta_i s^{-\mu_i} + \lambda \right) F(s). \tag{23}$$

5.1.3. Derivative representation

Using the equivalent form (19), the operator can be interpreted as a derivative-based structure:

$$\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f(t) = \frac{1}{2} q_1(t, \overline{\mu}) * \left(\frac{dq_2(t, \overline{\mu})}{dt} * \sum_{i=1}^{N} \theta_i I_t^{1-\mu_i} f(t) + q_2(t, \overline{\mu}) * \sum_{i=1}^{N} \theta_i D_t^{\mu_i} f(t) \right) + \lambda f(t). \tag{24}$$

Using (21), its Laplace representation remains

$$\mathcal{L}\left\{ \mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f \right\}(s) = \left(\frac{Q_1(s, \overline{\mu}) Q_2(s, \overline{\mu})}{2} \sum_{i=1}^{N} \theta_i s^{\mu_i} + \lambda \right) F(s). \tag{25}$$

This form highlights the direct contribution of the fractional derivatives.

5.2. PID-type structures generated by the nested weighted operator

The operator introduced in (18) naturally generates multi-memory integral and derivative structures through the differentiation of the nested fractional integrals. However, the simultaneous presence of independent integral and derivative channels is not directly obtained from the original definition. Nevertheless, motivated by the same weighted convolutional architecture, it is natural to consider the following extension:

Definition 9. Assume that $\overline{\mu} = (\mu_1, \dots, \mu_{N_1})$ and $\overline{\nu} = (\nu_1, \dots, \nu_{N_2})$ are finite sets of fractional orders with $0 < \mu_i, \nu_j < 1$, and let $\theta_i, \kappa_j, \lambda \in \mathbb{R}$. Moreover, assume that the weighting kernels $q_1(\cdot, \overline{\mu})$ and $q_2(\cdot, \overline{\mu})$ belong to $AC([0,T]) \cap L^1(0,T)$, and that $f \in AC([0,T]) \cap L^1(0,T)$ is such that all fractional integrals ${I_t}^{1-\mu_i} f(t)$ and ${I_t}^{1-\nu_j} f(t)$ exist. Finally, we assume that all convolution products involved in the definition are well-defined.

$$\mathcal{W}^{\boldsymbol{\mu},\boldsymbol{\nu}}_{t,\mathrm{conv}} f(t) = q_1(t,\overline{\mu}) * \left(\frac{d}{dt}\left(q_2(t,\overline{\mu}) * \sum_{i=1}^{N_1} \theta_i {I_t}^{1-\mu_i} f(t) \right) + \sum_{j=1}^{N_2} \kappa_j {I_t}^{1-\nu_j} f(t) \right) + \lambda f(t). \tag{26}$$

This extended operator preserves the nested weighted structure while introducing an additional independent integral.

Applying the derivative inside the first nested block, we obtain

$$\mathcal{W}^{\boldsymbol{\mu},\boldsymbol{\nu}}_{t,\mathrm{conv}} f(t) = q_1(t,\overline{\mu}) * \left(q_2(t,\overline{\mu}) * \sum_{i=1}^{N_1} \theta_i {D_t}^{\mu_i} f(t) + \sum_{j=1}^{N_2} \kappa_j {I_t}^{1-\nu_j} f(t) \right) + \lambda f(t). \tag{27}$$

This representation combines derivative and integral memory effects in a unified weighted framework, providing a natural mixed fractional PID-type structure. Its Laplace transform is

$$\mathcal{L}\left\{ \mathcal{W}^{\boldsymbol{\mu},\boldsymbol{\nu}}_{t,\mathrm{conv}} f \right\}(s) = \left(Q_1(s) Q_2(s) \left[\sum_{i=1}^{N_1} \theta_i s^{\mu_i} + \sum_{j=1}^{N_2} \kappa_j s^{-(1-\nu_j)} \right] + \lambda \right) F(s). \tag{28}$$

This representation combines proportional, derivative, and integral memory effects.

5.3. Illustrative examples

We now present several examples showing how the proposed weighted nested framework naturally generates different fractional PID-type structures.

Example 10 (*PD controller*).

Consider the simplest case $N = 2$, $q_1(t) = 1$, $q_2(t) = q$, with $q \in \mathbb{R}$ constant, and let $\mu_1 = \mu$, $\mu_2 = 0$, under the appropriate hypotheses for f. Then the operator (18) becomes

$$\mathcal{W}^{\mu,1}_t f(t) = \frac{d}{dt}\left(q\left(\theta_1 {I_t}^{1-\mu} + \theta_2 I_t \right) f(t) \right) + \lambda f(t).$$

Its Laplace transform is

$$\mathcal{L}\{\mathcal{W}^{\mu,1}_t f\}(s) = \left(q\beta_1 s^{\mu} + q\beta_2 + \lambda \right) F(s),$$

which corresponds to a Podlubny-type fractional PD controller.

Example 11 (*Memory Correction Term*). Product version Let $q_1(t) = 1$, $q_2(t) = 1 + t$, and consider $N = 2$, under the appropriate hypotheses for f. Then

$$\mathcal{W}^{\mu_1,\mu_2}_t f(t) = \frac{d}{dt}\left((1+t)(\beta_1 {I_t}^{1-\mu_1} + \beta_2 {I_t}^{1-\mu_2}) f(t) \right) + \lambda f(t).$$

Applying product rule yields

$$\mathcal{W}^{\mu_1,\mu_2}_t f(t) = (1+t)(\beta_1 {D_t}^{\mu_1} + \beta_2 {D_t}^{\mu_2}) f(t) + (\beta_1 {I_t}^{1-\mu_1} + \beta_2 {I_t}^{1-\mu_2}) f(t) + \lambda f(t).$$

- To illustrate the behavior of the proposed operator, consider $f(t) = \sin(2t)$, $t \in [0, 10]$, and the weighted fractional structure

$$\mathcal{W}^{\mu_1,\mu_2}_t f(t) = (1+t)\left(\beta_1 {D_t}^{\mu_1} + \beta_2 {D_t}^{\mu_2} \right) f(t) + \left(\beta_1 {I_t}^{1-\mu_1} + \beta_2 {I_t}^{1-\mu_2} \right) f(t) + \lambda f(t).$$

  We choose the parameters $\mu_1 = 0.3$, $\mu_2 = 0.7$, $\beta_1 = 0.78$, $\beta_2 = 0.2$, $\lambda = 0.5$. Thus, the operator becomes

$$\mathcal{W}^{\mu_1,\mu_2}_t f(t) = (1+t)\left(0.78 {D_t}^{0.3} + 0.2 {D_t}^{0.7} \right) \sin(2t) + \left(0.78 {I_t}^{0.7} + 0.2 {I_t}^{0.3} \right) \sin(2t) + 0.5 \sin(2t).$$

  Fig. 2 illustrates the response of the proposed nested weighted operator to a sinusoidal input. In contrast to conventional fractional operators, the resulting signal is shaped not only by the distributed-memory effects of the fractional derivatives and integrals but also by the time-varying weighting function $(1 + t)$. Consequently, the output exhibits a gradual amplitude modulation while preserving its oscillatory nature. The nested integral contribution further smooths the response and slightly modifies its phase, illustrating how the simultaneous selection of the fractional orders and weighting functions provides additional flexibility for shaping the operator dynamics. The numerical results were obtained using the Grünwald–Letnikov approximation for the fractional derivatives together with the Liouville representation for the fractional integrals.
- Consider the input signal

$$f(t) = e^{-t}, \qquad t \in [0, 10],$$

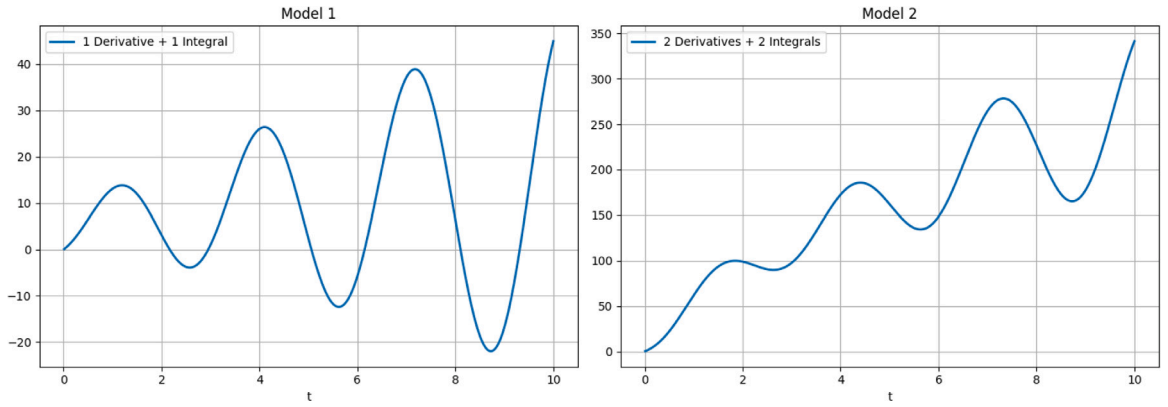


Fig. 2. Response of the proposed nested weighted fractional operator for the input $f(t) = \sin(2t)$. The left figure (Model 1) uses one Liouville fractional integral and one fractional derivative, whereas the right figure (Model 2) employs two Liouville fractional integrals and two fractional derivatives, providing a richer multi-memory dynamics.

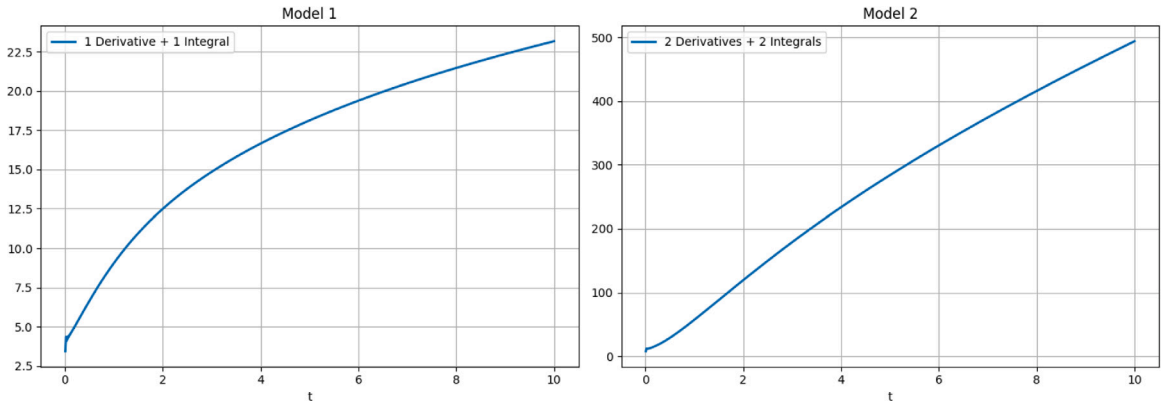


Fig. 3. Response of the proposed nested weighted fractional operator for the input $f(t) = e^{-t}$. The left figure (Model 1) uses one Liouville fractional integral and one fractional derivative, whereas the right figure (Model 2) employs two Liouville fractional integrals and two fractional derivatives, providing a richer multi-memory dynamics.

and the weighted fractional operator

$$\mathcal{W}_t^{\mu_1,\mu_2} f(t) = (1+t)\left(\beta_1 D_t^{\mu_1} + \beta_2 D_t^{\mu_2}\right) f(t) + \left(\beta_1 I_t^{1-\mu_1} + \beta_2 I_t^{1-\mu_2}\right) f(t) + \lambda f(t).$$

For the parameter values $\mu_1 = 0.3$, $\mu_2 = 0.7$, $\beta_1 = 0.78$, $\beta_2 = 0.2$, $\lambda = 0.5$, the operator becomes

$$\mathcal{W}_t^{\mu_1,\mu_2} f(t) = (1+t)\left(0.78 D_t^{0.3} + 0.2 D_t^{0.7}\right) e^{-t} + \left(0.78 I_t^{0.7} + 0.2 I_t^{0.3}\right) e^{-t} + 0.5 e^{-t}.$$

Fig. 3 illustrates the response of the proposed nested weighted operator for an exponentially decaying input. Although the excitation decreases monotonically with time, the combined action of the fractional derivatives, fractional integrals, and the time-varying weighting function produces a response that differs markedly from the original exponential profile. The distributed-memory property introduced by the fractional terms reshapes the transient dynamics, while the weighting function $(1+t)$ progressively modulates the operator response.

The obtained result demonstrates that the proposed formulation can effectively modify the temporal evolution of exponentially decaying signals while preserving their asymptotic behavior. More importantly, together with the previous sinusoidal example, it illustrates the generality of the proposed nested weighted operator. By jointly selecting the fractional orders and the weighting functions, the proposed framework provides additional degrees of freedom for tailoring the response of both oscillatory and monotonic signals, extending the capabilities of conventional fractional operators.

Remark 12 (***Memory Correction Term***)**.** The test functions $f(t) = \sin(2t)$ and $f(t) = e^{-t}$ are selected as representative benchmark examples due to their complementary analytical and dynamical properties. The trigonometric function $\sin(2t)$ provides an oscillatory signal, allowing us to evaluate how the proposed operator captures periodic behavior, phase effects, and memory interactions under convolutional structures. On the other hand, the exponential function e^{-t} represents a decaying process, which is particularly relevant

in the study of dissipative systems, transient responses, and stability analysis. Moreover, both functions possess well-known fractional derivatives and Laplace transforms, making them convenient prototypes for validating the theoretical properties of the proposed framework and illustrating its computational tractability.

Convolution version Let $q_1(t) = \delta(t)$, $q_2(t) = 1 + t$, and consider $N = 1$, under the appropriate hypotheses for f. Then

$$\mathcal{W}^{\mu_1,\mu_2}_{t,\mathrm{conv}} f(t) = \frac{d}{dt}\left((1+t) * (\beta_1 I_t^{1-\mu_1} + \beta_2 I_t^{1-\mu_2}) f(t)\right) + \lambda f(t).$$

Applying Proposition 1 yields

$$\mathcal{W}^{\mu_1,\mu_2}_{t,\mathrm{conv}} f(t) = (1+t) * D_t^{\mu} f(t) + \lambda f(t).$$

or

$$\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f(t) = 1 * I_t^{1-\mu} f(t) + \lambda f(t) = \int_a^t I_\tau^{1-\mu} f(\tau)\, d\tau + \lambda f(t).$$

On the other hand, applying (14) yields

$$\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f(t) = \frac{1}{2}(1+t) * D_t^{\mu} f(t) + \frac{1}{2} * I_t^{1-\mu} f(t) + \lambda f(t).$$

Its Laplace transform is

$$\mathcal{L}\{\mathcal{W}^{\boldsymbol{\mu}}_{t,\mathrm{conv}} f\}(s) = \frac{1}{2}\left(\frac{1}{s} + \frac{1}{s^2}\right) s^{\mu} F(s) + \frac{1}{2} s^{\mu-1} F(s) + \lambda F(s) = \frac{1}{2} s^{\mu-2} F(s) + s^{\mu-1} F(s) + \lambda F(s). \tag{29}$$

Example 12 (*Hierarchical Fractional Integral Structure*). Let $t \in [t_0, T]$, and assume $0 < \mu < 1, \alpha > 0, q \in C^1([t_0, T]), f \in L^1([t_0, T])$. Assume, in addition, that the fractional derivative $D_t^{\mu} f$ exists in the sense adopted throughout this work. Let $\lambda, \theta \in \mathbb{R}$. Consider the operator

$$\mathcal{W}_t^{\boldsymbol{\mu}} f = \partial_t\left[q(t) I_t^{1-\mu} + \lambda I_t\right] f + \theta I_t^{\alpha} f.$$

Using the product rule and the relation $\partial_t I_t^{1-\mu} f = D_t^{\mu} f$, the operator can be written as

$$\mathcal{W}_t^{\boldsymbol{\mu}} f = q(t) D_t^{\mu} f + \dot{q}(t) I_t^{1-\mu} f + \theta I_t^{\alpha} f + \lambda f.$$

We now analyze the hierarchy between the two fractional integral terms. Since $0 < \mu < 1$ and $\alpha > 0$, both $1 - \mu$ and α are positive, and the semigroup property

$$I_t^a I_t^b = I_t^{a+b}, \qquad a, b > 0,$$

can be employed.

Case 1: $\alpha > 1 - \mu$**.** Let

$$\alpha = (1-\mu) + A, \qquad A > 0.$$

Then, by the semigroup property,

$$I_t^{\alpha} = I_t^A I_t^{1-\mu},$$

and hence

$$\mathcal{W}_t^{\boldsymbol{\mu}} f = q(t) D_t^{\mu} f + \dot{q}(t) I_t^{1-\mu} f + \theta I_t^A I_t^{1-\mu} f + \lambda f. \tag{30}$$

Thus, the fractional integral $I_t^{\alpha} f$ contains $I_t^{1-\mu} f$ as an inner memory layer followed by an additional integration of order A.

Case 2: $0 < \alpha < 1 - \mu$**.** Let

$$1 - \mu = \alpha + A, \qquad A > 0.$$

Then

$$I_t^{1-\mu} = I_t^A I_t^{\alpha},$$

and consequently

$$\mathcal{W}_t^{\boldsymbol{\mu}} f = q(t) D_t^{\mu} f + \dot{q}(t) I_t^A I_t^{\alpha} f + \theta I_t^{\alpha} f + \lambda f. \tag{31}$$

In this case, $I_t^{1-\mu} f$ contains $I_t^{\alpha} f$ as an inner memory layer followed by an additional integration of order A.

Therefore, the relative ordering of α and $1 - \mu$ determines the nesting of the fractional integral operators and, consequently, the hierarchy of the associated memory layers.

Example 13 (***Hierarchical Fractional Derivative Structure***)**.** Let $t \in [t_0, T]$, and assume $0 < \alpha < 1, 0 < \mu < 1, q \in C^1([t_0, T]), f \in L^1([t_0, T])$. Assume that the fractional derivatives $D_t^\alpha f$ and $D_t^\mu f$ exist in the sense adopted throughout this work. Furthermore, assume the additional regularity required for the composition of fractional derivatives used below. Let $\lambda, \theta \in \mathbb{R}$. Consider the operator

$$\mathcal{W}_t^{\alpha,\mu} f = \partial_t \left[\theta I_t^{1-\alpha} + q(t) I_t^{1-\mu} + \lambda I_t \right] f.$$

Using the product rule and the identities $\partial_t I_t^{1-\alpha} f = D_t^\alpha f, \partial_t I_t^{1-\mu} f = D_t^\mu f$, we obtain

$$\mathcal{W}_t^{\alpha,\mu} f = q(t) D_t^\mu f + \dot{q}(t) I_t^{1-\mu} f + \theta D_t^\alpha f + \lambda f.$$

We now analyze the hierarchy between the two fractional derivative terms. Under the assumed composition property

$$D_t^{a+b} f = D_t^a D_t^b f,$$

the ordering of α and μ determines the corresponding nested derivative structure.

Case 1: $\alpha > \mu$**.** Let

$$\alpha = \mu + A, \qquad A > 0.$$

Then

$$D_t^\alpha = D_t^A D_t^\mu,$$

and therefore

$$\mathcal{W}_t^{\alpha,\mu} f = q(t) D_t^\mu f + \dot{q}(t) I_t^{1-\mu} f + \theta D_t^A D_t^\mu f + \lambda f. \tag{32}$$

Thus, the higher-order fractional derivative $D_t^\alpha f$ can be interpreted as an additional differentiation layer acting on the lower-order fractional derivative $D_t^\mu f$.

Case 2: $0 < \alpha < \mu$**.** Let

$$\mu = \alpha + A, \qquad A > 0.$$

Then

$$D_t^\mu = D_t^A D_t^\alpha,$$

and consequently

$$\mathcal{W}_t^{\alpha,\mu} f = q(t) D_t^A D_t^\alpha f + \dot{q}(t) I_t^{1-\mu} f + \theta D_t^\alpha f + \lambda f. \tag{33}$$

In this case, the lower-order fractional derivative $D_t^\alpha f$ constitutes the inner derivative layer, while $D_t^\mu f$ corresponds to an additional differentiation of order A.

Therefore, under the assumed composition property, the relative ordering of α and μ determines the nesting of the fractional derivatives and the corresponding hierarchical differential structure.

Remark 13 (***Computational Implementation and Finite-Memory Approximation***)**.** The nested fractional structure generally requires a larger history buffer than a conventional FOPID controller, since each fractional memory layer may involve its own history of past samples. Consequently, the computational and memory requirements increase with the number of fractional memory terms N. For real-time implementation, a practical approach is to employ a finite-memory approximation, in which the fractional history is truncated to a moving window of length T_m. Thus, only the most recent samples within $[t - T_m, t]$ are retained. The memory length T_m can be selected according to the desired accuracy and available computational resources, with larger values providing a more accurate approximation of the long-memory dynamics.

6. Constructive weight selection methodology

Once the generalized controller structure has been established, the remaining design problem consists of prescribing the weighting functions. Unlike the conventional FOPID controller, where the design variables are limited to controller gains and fractional orders, the proposed framework introduces the weighting functions as additional structural design parameters. Since different weight selections may produce comparable closed-loop responses, the objective is not to identify a unique solution but to develop a constructive design methodology that satisfies both the analytical requirements of the proposed operators and the practical constraints associated with controller implementation.

The first requirement is *analytical consistency*. In order for the weighted fractional operator to be well defined, the weighting functions must satisfy the regularity assumptions established throughout this paper. In particular, we assume

$$Q_1, Q_2 \in AC([0, T]) \cap L^1(0, T),$$

which guarantees the existence of the required convolution products and preserves the validity of the corresponding Laplace-domain representation.

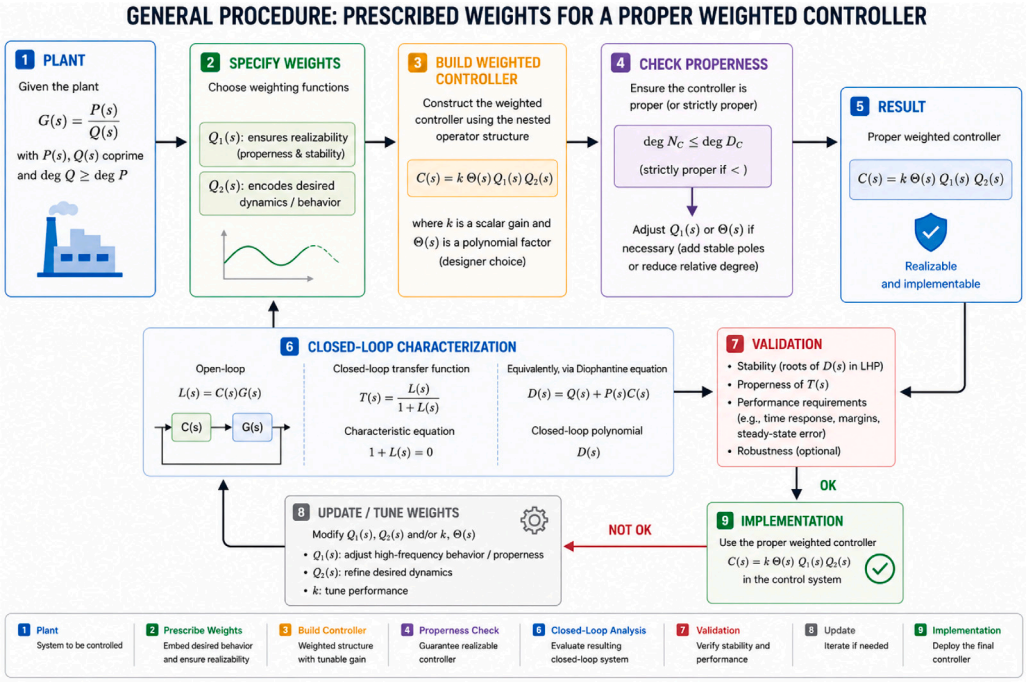


Fig. 4. Constructive methodology for prescribing the weighting functions.

The second requirement is *dynamic consistency*. The main idea is to prescribe the weighting functions so that the desired transient characteristics are embedded directly into the operator itself. Instead of constructing a desired closed-loop characteristic polynomial explicitly, the dominant dynamics are incorporated through the poles of the weighting functions.

The rationale behind prescribing the poles of $q_2(s)$ is that they directly determine the dominant memory dynamics introduced by the weighted operator. Consequently, transient specifications can be incorporated into the controller through the weighting structure itself before the closed-loop characteristic polynomial is obtained. In this manner, the weighting functions become additional structural design variables that complement the conventional controller parameters.

Based on these principles, the weighting functions can be prescribed according to the following constructive design procedure:

1. Specify the desired dominant closed-loop dynamics.
2. Select $q_2(s)$ so that its poles reproduce the desired dominant dynamics.
3. Choose $q_1(s)$ to guarantee realizability and properness of the resulting controller.
4. Construct the controller through the proposed nested weighted operator representation.

This methodology provides a systematic design procedure in which the weighting functions are prescribed to achieve the desired dynamic behavior while preserving the analytical well-posedness and realizability of the resulting fractional operator. As illustrated in Fig. 4, the design process begins with the plant model, from which the weighting functions are selected according to the desired closed-loop characteristics and incorporated into the nested operator to construct the controller. The resulting controller is then verified for properness, analyzed in closed loop, and validated with respect to stability, performance, and robustness. If the prescribed specifications are not satisfied, the weighting functions and controller parameters are iteratively refined until the desired behavior is achieved.

The proposed workflow emphasizes that the weighting functions are fundamental design variables rather than simple scaling factors. Their appropriate selection simultaneously guarantees operator realizability and provides additional degrees of freedom for shaping the closed-loop dynamics, extending the capabilities of conventional fractional-order controller design.

Example 14 (***Prescribed Fractional Weights and Emergent Dominant Modes***).
Consider the plant

$$G(s) = \frac{1}{(s+2)(s+3)}.$$

The proposed weighted fractional controller is constructed from the same structural form introduced in Section 6, namely

$$C_F(s) = k\,\Theta(s)\,Q_1(s)\,Q_2(s),$$

where $\Theta(s) = s^{\mu} + 1$, $0 < \mu < 1$, and $Q_1(s) = \dfrac{s^{\beta} + 5}{s^{\beta} + 10}$, $0 < \beta < 1$, while $Q_2(s) = \dfrac{1}{s^{2\alpha} + 4.2s^{\alpha} + 9}$, $0 < \alpha < 1$. Hence, the fractional controller is given by

$$C_F(s) = k\,\frac{(s^{\mu} + 1)(s^{\beta} + 5)}{(s^{\beta} + 10)(s^{2\alpha} + 4.2s^{\alpha} + 9)}.$$

For the numerical study, the parameters are selected as

$$k = 1, \qquad \alpha = 0.7, \qquad \beta = 0.5, \qquad \mu = 0.6.$$

These values satisfy $0 < \alpha < 1$, $0 < \beta < 1$, $0 < \mu < 1$, as required for the fractional-order formulation. Moreover, $2\alpha = 1.4 > \mu = 0.6$, which guarantees a strictly proper high-frequency behavior of the fractional controller. Indeed,

$$C_F(s) \sim k s^{\mu - 2\alpha}$$

as $|s| \to \infty$, and therefore $C_F(s) \to 0$ for the selected orders. This provides a realizable controller structure from the high-frequency properness viewpoint.

The fractional weighting function $Q_2(s)$ is selected to introduce a prescribed fractional modal structure. Its characteristic equation is

$$s^{2\alpha} + 4.2s^{\alpha} + 9 = 0.$$

Introducing the auxiliary variable $z = s^{\alpha}$ yields $z^2 + 4.2z + 9 = 0$, whose roots are $z_{1,2} = -2.1 \pm 2.14\,\mathrm{i}$. Accordingly, the associated fractional modes are $s_{1,2} = (-2.1 \pm 2.14\,\mathrm{i})^{1/\alpha}$, with the appropriate branch convention for the fractional power.

These fractional modes should not be interpreted directly as the poles of the complete closed-loop system. Instead, they characterize the fractional modal structure introduced through $Q_2(s)$ and therefore contribute to the shaping of the transient response. The actual closed-loop modes are determined by the complete characteristic equation $1 + C_F(s)G(s) = 0$. The remaining weighting function $Q_1(s)$ is proper and satisfies $\lim_{|s|\to\infty} Q_1(s) = 1$. Consequently, it does not compromise the properness of the overall controller and provides an additional degree of freedom for fractional transient shaping.

For comparison, an integer-order benchmark is constructed directly from the same controller structure by setting all fractional orders equal to one, $\alpha = \beta = \mu = 1$. Thus, with the same gain $k = 1$, the corresponding integer-order controller is

$$C_I(s) = \frac{(s + 1)(s + 5)}{(s + 10)(s^2 + 4.2s + 9)}.$$

This choice provides a structurally consistent benchmark: the integer-order and fractional-order controllers differ only through the fractional realization of the corresponding powers of s. Hence, the comparison does not rely on independently tuned controller structures, but rather contrasts the integer-order realization with its fractional counterpart.

The fractional powers appearing in $C_F(s)$ are implemented using the Oustaloup recursive approximation. Specifically, an Oustaloup recursive approximation with $N = 3$ is employed for each fractional power, with

$$\omega_b = 0.1 \text{ rad/s}, \qquad \omega_h = 10 \text{ rad/s}.$$

Thus, s^{μ}, s^{β}, and s^{α} are replaced by their corresponding rational approximations over the frequency interval $0.1 \le \omega \le 10$ rad/s. The resulting rational approximation provides a finite-dimensional realization suitable for numerical simulation and implementation.

For both controllers, the open-loop transfer function is defined as

$$L(s) = C(s)G(s),$$

where $C(s) = C_I(s)$ for the integer-order benchmark and $C(s) = C_F(s)$ for the proposed fractional controller. The corresponding closed-loop transfer function is

$$T(s) = \frac{L(s)}{1 + L(s)}.$$

The integer-order benchmark has the plant poles $s = -2$, $s = -3$, while the controller introduces the poles associated with $s = -10$ and $s^2 + 4.2s + 9 = 0$.

No exact pole–zero cancellation between the plant and controller is introduced by construction. In particular, the plant poles -2 and -3 do not coincide with any controller zeros. For the integer-order closed loop, the actual poles are obtained from the complete characteristic equation $1 + C_I(s)G(s) = 0$, rather than from the controller or plant poles considered independently.

For the fractional controller, the same principle applies. The fractional modes introduced by $Q_2(s)$ interact with the plant dynamics through the complete closed-loop characteristic equation $1 + C_F(s)G(s) = 0$. Therefore, possible unfavorable interactions between the plant dynamics and the fractional weighting modes are not inferred from $Q_2(s)$ alone, but are assessed from the resulting closed-loop system. For the parameter set considered here, the resulting closed-loop response is asymptotically stable.

Fig. 5 compares the impulse responses obtained with the integer-order benchmark controller and the proposed fractional controller. Both responses decay toward zero, confirming the asymptotically stable behavior observed for the selected parameter sets. Nevertheless, their transient characteristics are markedly different.

The integer-order controller exhibits a pronounced initial peak, followed by a noticeable undershoot before converging to the equilibrium. In contrast, the proposed fractional controller produces a substantially smaller initial peak, a reduced undershoot,

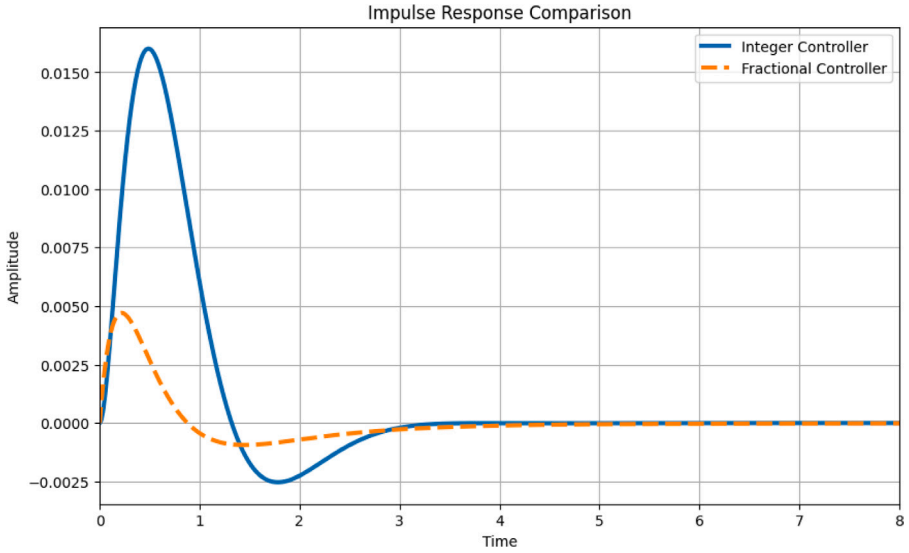


Fig. 5. Impulse response comparison between the integer-order benchmark controller and the proposed nested weighted fractional controller.

and a smoother decay toward the steady state. This behavior is consistent with the distributed-memory effect introduced by the fractional weighting structure, which provides additional degrees of freedom for shaping the transient response. In particular, the fractional weighting does not simply add additional poles to the controller. Rather, it modifies the distribution of the dynamic memory across different fractional orders. The interaction of these memory components with the plant dynamics determines the resulting closed-loop transient behavior.

Therefore, this example illustrates how the proposed nested weighted formulation extends the classical pole-shaping philosophy to the fractional-order setting. Instead of prescribing the complete closed-loop characteristic polynomial directly, fractional modal information is embedded in the weighting structure, while the actual closed-loop dynamics, realizability, and stability are assessed at the level of the complete controller–plant interconnection.

7. Conclusions

This paper developed a unified nested weighted framework for the systematic construction of generalized fractional operators and their application to Podlubny-type fractional PID controllers. Unlike conventional formulations that rely on predefined fractional operators, the proposed methodology adopts a composition-based perspective in which generalized operators are generated through recursive combinations of weighted differential and integral mappings. From this viewpoint, the classical $PI^{\lambda}D^{\mu}$ controller naturally emerges as a particular case within a broader family of weighted fractional PID controllers.

A central contribution of this work is the establishment of a constructive connection between generalized fractional operator theory and controller synthesis. Within the proposed framework, the introduced pulling-type weighted operator was shown to satisfy the fundamental analytical properties required for fractional-order modeling, including linearity, kernel characterization, boundedness, semigroup behavior, and Laplace-domain representation. These results provide the mathematical foundation for constructing generalized controller families while preserving the physical interpretation of the classical Podlubny formulation.

The proposed framework also provides a new interpretation of fractional PID design. Rather than selecting a particular fractional operator a priori, the controller is obtained through the systematic construction of nested weighted operators, where the weighting functions become structural design variables governing the memory distribution and the resulting dynamic behavior. Consequently, operator construction and controller synthesis are unified within a common analytical framework.

Future research will focus on extending the proposed framework to broader classes of generalized fractional operators, developing systematic methodologies for weighting-function selection, investigating adaptive and variable-order extensions, and exploring applications to nonlinear, distributed-parameter, and multi-agent dynamical systems.

CRediT authorship contribution statement

Antonio Di Teodoro: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Juan J. Gude:** Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Oscar Camacho:** Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Universidad San Francisco de Quito through the Poli-Grants Program under Grant No. 41983. The authors also acknowledge the financial support provided by the Basque Government through the consolidated research group D4K–Deusto for Knowledge under Grants IT1528-22 and IT1796-26.

Data availability

No data was used for the research described in the article.

References

[1] Podlubny I. Fractional-order systems and PI/sup/spl lambda//D/sup/spl mu//-controllers. IEEE Trans Autom Control 2002;44(1):208–14.
[2] Monje CA, Chen Y, Vinagre BM, Xue D, Feliu V. Fractional-order systems and controls: fundamentals and applications. Springer; 2010.
[3] Meneses H, Arrieta O, Padula F, Vilanova R, Visioli A. PI/PID control design based on a fractional-order model for the process. IFAC-PapersOnLine 2019;52(1):976–81.
[4] Bashishtha T, Singh V, Yadav UK, Sahu UK. Fractional-order PID controllers and applications: A comprehensive survey. Annu Rev Control 2025;60:101013.
[5] Yokuş YE, Ayten UE, Kartcı A. A review of the state-of-the-art in fractional-order PID controllers. Fract Calc-Theory Appl 2024.
[6] Wendimu AA, Matu R, Shaikh I, Kebede ZK. Fractional-order PID control for elevation and azimuth in a twin rotor system. Sci Rep 2025;15(1):33590.
[7] Camacho O, Vega S, Herrera M, Di Teodoro A, Gude JJ. A fractional order pid-based sliding mode controller approach for chemical processes. Results Control Optim 2025;20:100592.
[8] Petráš I. Fractional-order controllers: A practical approach. IEEE Trans Syst Man Cybern: Syst 2025.
[9] Di Teodoro A, Gude JJ, Vega S, Chalco R, Montaluisa R, Camacho O. Fractional-order hybrid 2-DOF PID control with a complex basis: A novel framework for MIMO systems. Asian J Control 2026;28(1):57–75.
[10] Sorcia-Vázquez FdJ, Rumbo-Morales JY, Brizuela-Mendoza JA, Ortiz-Torres G, Sarmiento-Bustos E, Pérez-Vidal AF, Rentería-Vargas EM, De-la Torre M, Osorio-Sánchez R. Experimental validation of fractional PID controllers applied to a two-tank system. Mathematics 2023;11(12):2651.
[11] Valdez-Resendiz JE, Rumbo-Morales JY, Sorcia-Vázquez FD, Ortiz-Torres G, Gómez-Aguilar J, Cantero CAT, Ramos-Martinez M. Optimization and discrete fractional-order PID controller applied to a pressure swing adsorption plant for the production and recovery of bioethanol. J Clean Prod 2024;476:143618.
[12] Sorcia-Vázquez FD, Ortiz-Torres G, Rumbo-Morales JY, Villalobos AUS, Torres-Cantero CA, Ramos-Martinez M, Rosales AM, López-Osorio MA, Juarez MA, Beltrán-Escobar M, et al. Characterization and experimentation with natural and synthetic zeolites to be selective toward the water molecule: Bioethanol production from a robust control law applied to the complex process of pressure swing adsorption. South Afr J Chem Eng 2026;100881.
[13] Seo S-W, Choi HH. Digital implementation of fractional order PID-type controller for boost DC–DC converter. IEEE Access 2019;7:142652–62.
[14] Camacho O, Chalco R, Di Teodoro A, Gude JJ, Montaluisa R, Vargas C, Vega S. A weighted fractional order PID controller for nonlinear systems with variable delay. IFAC-PapersOnLine 2025;59(37):199–204.
[15] Fernandez A, Baleanu D, Srivastava HM. Series representations for fractional-calculus operators involving generalised Mittag-Leffler functions. Commun Nonlinear Sci Numer Simul 2019;67:517–27.
[16] Fernandez A, Fahad HM. Weighted fractional calculus: a general class of operators. Fractal Fract 2022;6(4):208.
[17] Fahad HM, Fernandez A, Rehman MU, Siddiqi M. Tempered and Hadamard-type fractional calculus with respect to functions. Mediterr J Math 2021;18(4):143.
[18] Camacho O, Chalco R, Di Teodoro A, Gude JJ, Montaluisa R, Vargas C, Vega S, Villegas M. An introduction to weighted operators via composition and selected properties, aimed at numerical implementation. IFAC-PapersOnLine 2025;59(37):47–52.
[19] Kilbas AA, Marichev O, Samko S. Fractional integrals and derivatives (theory and applications). Gordon and Breach, Switzerland; 1993.
[20] Kilbas AA, Srivastava HM, Trujillo JJ. Theory and applications of fractional differential equations, vol. 204, elsevier; 2006.
[21] Miller KS, Ross B. An introduction to the fractional calculus and fractional differential equations. Wiley; 1993.
[22] Podlubny I. Fractional differential equations. San Diego: Academic Press; 1999.
[23] Trujillo JJ, Scalas E, Diethelm K, Baleanu D. In: Fractional calculus: models and numerical methods, vol. 5, World Scientific; 2016.
[24] Cannarsa P, D'Aprile T. Introduction to measure theory and functional analysis. Springer; 2015.
[25] Folland GB. Real analysis: modern techniques and their applications. 2nd ed. New York: Wiley; 1999.
[26] Tao T. An introduction to measure theory. In: Graduate studies in mathematics, vol. 126, American Mathematical Society; 2011.
[27] Coloma N, Di Teodoro A, Ochoa-Tocachi D, Ponce F. Fractional elementary bicomplex functions in the Riemann–Liouville sense. Adv Appl Clifford Algebr 2021;31(4):63.
[28] Ceballos J, Coloma N, Di Teodoro A, Ochoa-Tocachi D. Generalized fractional Cauchy–Riemann operator associated with the fractional Cauchy–Riemann operator. Adv Appl Clifford Algebr 2020;30(5):1–22.
[29] Ceballos J, Coloma N, Di Teodoro A, Ochoa-Tocachi D, Ponce F. Fractional multicomplex polynomials. Complex Anal Oper Theory 2022;16(4):1–30.
[30] Diethelm K, Ford NJ. Analysis of fractional differential equations. J Math Anal Appl 2002;265(2):229–48.
[31] Caputo M. Linear models of dissipation whose Q is almost frequency independent—II. Geophys J Int 1967;13(5):529–39.
[32] Dyke PP. In: An introduction to Laplace transforms and Fourier series, vol. 517, London: Springer; 2001.
[33] Rudin W. Real and complex analysis. 3rd ed. McGraw-Hill; 1987.
[34] Widder DV. The Laplace transform. Princeton University Press; 1946.
[35] Yosida K. Functional analysis. 6th ed. Springer; 1980.
[36] Hilfer R. Applications of fractional calculus in physics. World scientific; 2000.
[37] Katugampola UN. A new approach to generalized fractional derivatives. Bull Math Anal Appl 2014;6(4):1–15.
[38] Petráš I. Fractional-order nonlinear systems: modeling, analysis and simulation. Springer Science & Business Media; 2011.
[39] Almeida R. A Caputo fractional derivative of a function with respect to another function. Commun Nonlinear Sci Numer Simul 2017;44:460–81. http://dx.doi.org/10.1016/j.cnsns.2016.09.006.
[40] Diethelm K, Ford NJ. Multi-order fractional differential equations and their numerical solution. Appl Math Comput 2004;154(3):621–40.

[41] Mainardi F. Fractional calculus and waves in linear viscoelasticity: an introduction to mathematical models. World Scientific; 2022.
[42] Almeida R, Malinowska AB, Odzijewicz T. Caputo–Katugampola fractional derivatives and applications. J Comput Phys 2016.
[43] Contreras E, Di Teodoro A, Mena M. Fractional Einstein field equations in $2+1$ dimensional spacetime. Gen Relativity Gravitation 2025;57(5):1–21.
[44] Katugampola UN. New fractional integral unifying six existing fractional integrals. 2017, arXiv preprint arXiv:1612.08596.
[45] Podlubny I. Fractional-order systems and fractional-order controllers. Inst Exp Phys Slovak Acad Sci Kosice 1994;12(3):1–18.
[46] Zhang B, Pi Y, Luo Y. Fractional order sliding-mode control based on parameters auto-tuning for velocity control of permanent magnet synchronous motor. ISA Trans 2012;51(5):649–56.
[47] Petráš I. Tuning and implementation methods for fractional-order controllers. Fract Calc Appl Anal 2012;15(2):282–303.