International Journal of Robotics and Control Systems
Vol. 5, No. 2, 2025, pp. 1211-1237
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Recent Advances in Energy-Efficient Fractional-Order PID Control for Industrial PLC-Based Automation: A Review

Sandra Francis [a,1], Pritesh Shah [a,2,*], Abhay Pal Singh [b,3], Ravi Sekhar [a,4]

[a] Symbiosis Institute of Technology, Symbiosis International (Deemed) University, Pune 412115, Maharashtra, India
[b] Department of Mechanical Engineering and Technology Management, Norwegian University of Life Sciences (NMBU), Kajaveien 25, As, 1433, Norway
[1] francis.sandra.phd2023@sitpune.edu.in; [2] pritesh.shah@sitpune.edu.in; [3] abhaya.aps@gmail.com;
[4] ravi.sekhar@sitpune.edu.in
* Corresponding Author

ARTICLE INFO

Article History
Received March 08, 2025
Revised April 16, 2025
Accepted May 07, 2025

Keywords
PLC-Based Fractional Control;
Metaheuristic Optimization in Control;
Energy-Efficient Industrial Automation;
Adaptive Fractional Controllers;
Hardware in Loop Testing

ABSTRACT

Through intelligent control and data-driven decision-making, Industry 4.0 transforms industrial automation by combining the digital, physical, and virtual worlds. The use of advanced control techniques, especially Fractional-Order PID (FOPID) controllers, has drawn a lot of attention due to the rising need for accurate and energy-efficient industrial automation. By examining recent developments in the application of energy-efficient FOPID controllers for Programmable Logic Controller (PLC) based automation systems, this review tries to bridge a gap in the body of literature. The study thoroughly examines more than ten years of research, classifying contributions according to optimization, fractional calculus approximations, and control design techniques. The reported results from various studies are compared using key performance indicators like energy consumption, ISE, ITAE, and IAE. The results show that FOPID controllers continuously perform better than classical PID in terms of energy efficiency, robustness, and control accuracy. However, there are still difficulties in striking a balance between real-time constraints and computational complexity, particularly in industrial settings. This review emphasizes how FOPID controllers can be used to achieve automation that is Industry 4.0 compatible, adaptive, and energy-efficient. It also emphasizes the necessity of future studies into hybrid optimization and lightweight implementation for next-generation PLC systems, as well as the need for standardized benchmarking frameworks.

1. Introduction

Energy-efficient industrial automation is crucial in meeting global sustainability goals, yet conventional PID controllers in PLC-based systems often struggle to optimize energy consumption effectively. FOPID controllers, with their superior flexibility and robustness, present a promising alternative, but their integration into industrial PLCs remains underexplored. While global initiatives such as the Paris Agreement and the UN's Sustainable Development Goals (SDGs), specifically SDG 7 and SDG 13, emphasize the need for energy-efficient technologies, the direct role of FOPID in achieving these objectives is rarely examined. A bridge can be created by examining current developments in FOPID control for industrial automation, with a focus on energy efficiency, optimization strategies, and real-time implementation on PLCs.By identifying key challenges—such as the trade-off between computational complexity and real-time implementation, the effectiveness of metaheuristic tuning methods, and the role of Industry 4.0 in adaptive energy-efficient control—this review highlights critical research directions and sets the stage for future advancements in sustainable automation [1]. Fig. 1 presents the interrelationships between the system components described above.

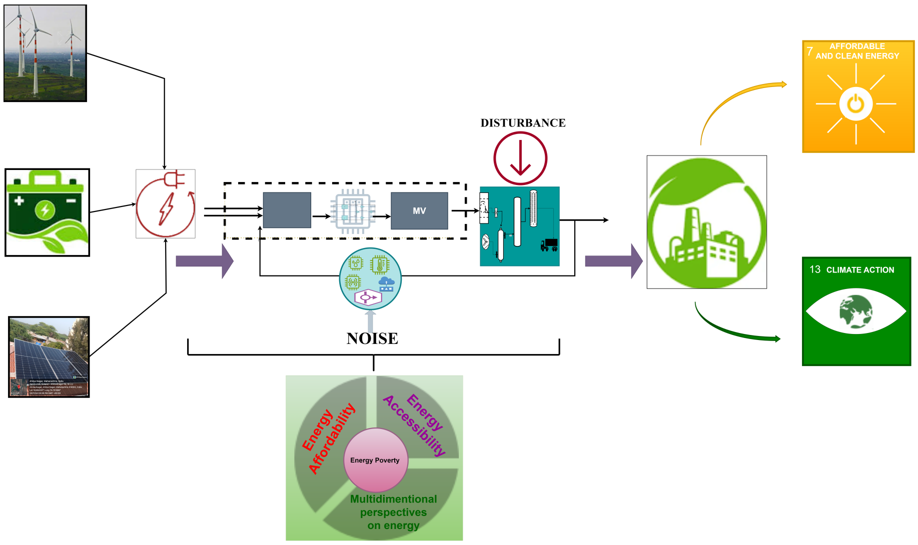

Fig. 1. Fractional-Order PID control in the context of SDG goals

1.1. Background and Motivation

The first reference of fractional orders was associated with Leibniz and L'Hˆopital in 1695 where a half-order derivative was mentioned [2].Over the years, questions have arisen as to why there is a need to transition to a fractional counterpart if present traditional methods are already there. What role does PLC [3] play in predicting performance gain, best tuning methods, minimum experiments to tune controller, and how to design a controller best have been tried to be addressed by many stalwarts to some extent working in this field [4]. The theory of fractional calculus is the generalization of integration and differentiation to non-integer order fundamental operator ${}_aD_t^\alpha$ where a and t are the bounds of the operator. It was introduced during the late 19th to Early 20th Century, which underlies these controllers. The continuous fractional integrodifferential operator is defined as shown in Equation (1) where for $a < 0$ it acts as an integrator and for $a > 0$ as a differentiator [5].

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

The evolution of nonlinear control systems such as robotic manipulators with coupled dynamics, chemical processes with exothermic reactions, power systems, and biological systems with complex feedback loops from the 18th century, when the centrifugal governor was invented, was propelled by the advancements made by Lyapunov and the integration of mechanics with feedback theory in the 1960s [6]. Practical Implementation from the 90s till today due to the advances in digital computation and control theory facilitated the practical application of fractional order controllers. The work of scientists like Igor Podlubny in 1994 [7] considered a simple unity feedback control system as shown in Fig. 2. is commendable.

Fractional-order transfer functions, whose simplest form in the Laplace domain is s^{α} , where s is the Laplace transform variable and $\alpha \in \mathbb{R}$ can be used to define fractional-order dynamic systems. The mathematical framework being fractional calculus, FOPID allows for a more nuanced representation of system dynamics. This capability allows us to model processes with memory and hereditary effects.

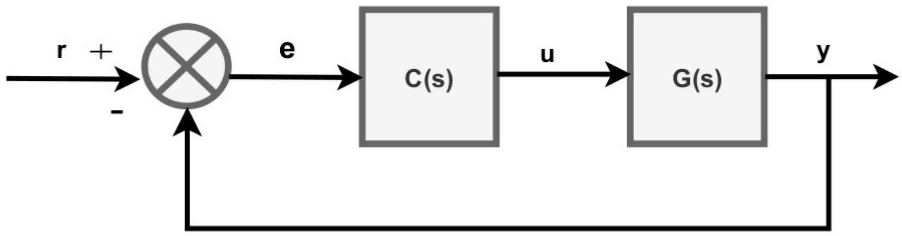

Fig. 2. Closed-Loop unity feedback system

These controllers can be combined with other techniques, such as neural networks, to further enhance their ability to manage nonlinearity. Advanced optimization techniques, such as modified harmony search algorithms, can be employed alongside these controllers to optimize their performance in nonlinear environments. Hybrid optimization approaches, which integrate the strengths of multiple algorithms (e.g., Genetic Algorithm (GA) [57]–[60] Particle Swarm Optimization (PSO) [61] or metaheuristics enhanced by artificial intelligence), remain under-explored. There is potential for enhancement of the outcomes derived from these optimization methods by suitable representation selection, parametrization, and methodology. An ideal and desired performance can be attained by comparing the outcomes of all the algorithms, including the Grey Wolf Optimization algorithm (GWO) and the Artificial Ecosystem Optimization algorithm (AEO) [8]. These methods could improve convergence speed, robustness, and global search capability, particularly for highly nonlinear and high-dimensional systems. Stated differently, fractional calculus permits a trade-off between a differentiator's high-frequency gain and an integrator's phase lag. Together, PLCs and FOPID controllers form a powerful synergy in industrial automation, enabling adaptive and precise control for complex, dynamic processes that became central to process control. Some simple plug-and-play types of methods that satisfy most of the systems are the need of the hour [2]. The paper aims to include contributions that are a comprehensive synthesis of approximation strategies for PLC-based FOPID control, which are available, comparative evaluation of energy-efficient optimization techniques, a discussion on practical trade-offs between computational complexity and control performance and insights into AI-driven and Industry 4.0-enabled adaptive FOPID control for industrial automation.

1.2. Role of Advanced Control Techniques

The increasing demand for energy efficiency in industrial automation has led to the adoption of advanced control techniques that optimize performance while reducing power consumption. Advanced control techniques in the context of PLCs enable precise, adaptive, and optimized control strategies, reducing energy consumption while maintaining operational stability. As an advanced control approach, these controllers offer superior flexibility and robustness compared to traditional integer-order controllers. The adoption of this technique improves dynamic response, reduces steady-state errors, and enhances disturbance rejection, leading to more energy-efficient industrial processes. The ability of such systems to handle complex system dynamics with fewer resources makes them particularly suitable for PLC-based automation.

Moreover, advanced tuning methods, optimization techniques, and real-time implementation strategies further refine their effectiveness in energy-efficient applications.FOPID controllers, based on fractional calculus, offer greater flexibility and accuracy in system control compared to traditional integer-order controllers by utilizing non-integer derivatives, allowing finer control over transient and steady-state performance. This results in improved dynamic response, as they better adapt to system variations, reducing overshoot and settling time, which minimizes energy losses. Additionally, they enhance robustness by providing better disturbance rejection and stability, ensuring efficient energy usage even under fluctuating load conditions. Their optimized control actions, facilitated by additional tuning parameters, enable more precise control, reducing unnecessary actuator activity and power consumption. Integrating FOPIDs into PLCs, the backbone of industrial automation, requires advancements in both hardware and software capabilities. Efficient computational algorithms are essential for handling the complexity of fractional derivatives, necessitating real-time numerical approximations. Digital implementation techniques, including advancements in digital signal processing (DSP) and embedded systems, enable effective programming into PLC architectures.

Furthermore, adaptive and self-tuning control strategies, incorporating AI-driven techniques such as machine learning and heuristic optimization methods, enhance performance in energy-efficient industrial applications. This paper reviews advances in energy-efficient control techniques using FOPIDs in PLCs, highlighting their benefits over traditional integer-order controllers. It explores this integration for industrial automation, analyzing compatibility, programming methods, and execution efficiency. How FOPIDs contribute to energy savings by optimizing power consumption in control systems while enhancing stability, response time, and robustness. Additionally, case studies and performance analyses demonstrate the advantages in real-world applications. The paper also discusses key challenges, such as computational complexity and hardware limitations, and suggests future research directions to improve the efficiency and applicability of FOPIDs in industrial automation.

1.3. Structure of Paper

The paper follows a logical progression, starting with background information and fundamental concepts before delving into implementation techniques, challenges, and future research directions. The section-wise breakdown is as follows:

- Section 1 gives the background and motivation providing an introduction to the significance of energy-efficient control in industrial automation and the role of advanced control strategies in optimizing energy consumption
- Section 2 gives the mathematical foundations of FOPID controllers
- Section 3 gives an overview of PLCs, which are widely used in industrial automation for process control, machine automation, and system monitoring. It covers the architecture of PLCs, their working principles, and their advantages in industrial settings
- Section 4 provides the previous research summary aiming to identify gaps in the implementation of an energy-efficient fractional-order control strategy for PLC-based systems and contains data from peer-reviewed journals over the last 4-5 years.
- Section 5 explores methods for implementing fractional-order controllers (FOCs) in PLC-based systems, including numerical approximation techniques and real-time computational algorithms. It highlights key challenges, such as hardware limitations, computational complexity, and the need for efficient tuning methods.
- Section 6 discusses the importance of energy-efficient control strategies in industrial automation.
- The seventh section gives the conclusions and future trends.

The research contribution is the synergistic integration of FOPID control with energy aware multi objective optimization and hybrid metaheuristic tuning strategies that seamlessly can be executed in real-time on PLC platforms.This may help in yielding superior dynamic response, robustness, and process efficiency while directly advancing sustainable industrial automation through significant reductions in energy consumption and carbon emissions.

2. Fundamentals of Fractional Order Controllers

2.1. Introduction and Definition

Controllers are the ”brains” that enable all control systems to function properly [2]. For this purpose, Integer- Order Proportional integral derivative (IOPID), also known as the traditional or conventional Proportional Integral Derivative (PID) implemented on PLC has been used extensively in process industries. There are several limitations that are associated with these controllers, one of that are manual tuning due to the presence of fixed tuning parameters, making them less adaptive [70]. Due to the control actions that are sudden and problems of overshooting, the consumption of energy is greater. The focus is more on setpoint tracking rather than the management of energy consumption or efficiency [30], [31], [76]. Hence, to achieve the above goal, the FOPID controller also written as $PI^{\lambda}D^{\mu}$ [32] as shown in Fig. 3 was among the many innovations that emerged as a lucrative choice. It was built on the foundation of traditional PID controllers [4] incorporating fractional-order terms, denoted by λ and μ that are fractional (non-integer) orders of differentiation and integration that provide the flexibility to tailor responses that are system-specific. If the λ and μ are equal to 1, the FOPID reduces to the simple PID [8] controller. These terms provide an added degree of freedom and flexibility, allowing FOPID to better model and control complex systems with nonlinearity, uncertainty, or time delays that are generally present in industrial settings. These observations underscore the robustness and adaptability of fractional-order controllers using them in cascade configurations [69], [77] as well in some applications, making them advantageous for challenging control applications and performing better.

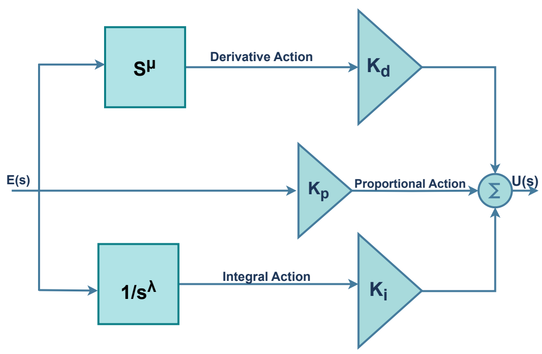

Fig. 3. Fractional-Order Proportional-Integral-Derivative (FOPID) control structure

The controller in generalized form is as shown in Equation (2), and in the time domain, it is given in Equation (3). The additional terms provide a greater degree of freedom belonging to the K_i and K_d gains. The ability to describe phenomena with memory makes them suitable for systems with non-local dynamics and long-term memory effects [23], [33].

$$C(s) = K_p + K_i/s^{\lambda} + K_d s^{\mu} \tag{2}$$

$$C(t) = K_p \left(e(t) + \frac{1}{T_I^{\lambda}} D^{-\lambda} e(t) + T_D^{\mu} D^{\mu} e(t) \right) \tag{3}$$

A fractional order between 0 and 1 can help achieve smoother control with reduced steady-state errors. The values of λ and μ are in the range of 0 and 1; combinations of these two parameters can be made to achieve certain goals. Fig. 2. shows the structure of a unity negative feedback control system. In this linear time-invariant control system, the plant and the controller are described by transfer functions G(s) and C(s), respectively. The reference input, the control signal, and the system output are r, u, and y, respectively. The advantage of fractional-order transfer functions is that they offer in-between qualities because their magnitude and phase response are not typical of integer-order transfer functions, hence the choice made is justified. The closed-loop transfer function in the control system of Fig (2). is given by Equation (4).

$$T(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \tag{4}$$

Using digital calculations, the FOPID controller governs various systems. The tuning of fractional order controllers is not limited to integer values, allowing for a more nuanced control strategy with a lot of background study on the different definitions of FOPID. When processes display non-local behavior, the FOPID controller performs better than the conventional PID controller and is also more energy efficient compared to IOPID [34]. PID controllers rely on integer order derivatives that oversimplify the behavior of any system under consideration. By leveraging the fractional-order parameters, FOPID controllers can adapt to changing environmental conditions (e.g., wind load or varying solar intensity) and adjust tracking operations accordingly, further reducing energy wastage. A wide class of functions that are used in control theory are associated with the Grünwald–Letnikov (GL), Riemann-Liouville (RL), and Caputo (C) definitions [49]. There are several types of approximation methods to realize the FOPID on PLC-like platforms:

- Numerical Approximation Methods like GL Approximation and RL approximation method.
- Frequency Domain Approximations
- Rational Approximation Methods like the Continuous Fraction Expansion (CFE) Method, Oustaloup's Recursive Approximation [45]–[47] Chareff's Method [48], [50]
- Integer-Order Equivalent Approximations
- Discretization Techniques like Tustin's Approximation (Bilinear Transformation), and Euler Approximation.
- Optimization-Based Tuning Methods Optimization is often used to tune parameters of the approximations to achieve the desired control performance under PLC constraints.

Optimization-based methods find the best set of parameters by minimizing the objective function as shown in Fig. 4 where the flow as to how the optimization is performed is depicted. Model-Based and Heuristic Approaches like CRONE (Commande Robuste d'Ordre Non Entier) Approach, Oustaloup's Recursive Approximation (ORA) [62]–[64]. Artificial Neural Networks (ANNs) etc. Table 1 and Table 2 present a comparison of various tuning methodologies based on key performance criteria.

2.2. Benefits Over Traditional Controllers

A control system is stable if its closed-loop transfer function meets the Nyquist stability criterion and the Bode phase margin requirements. In conventional PID controllers, stability margins depend on integer-order parameters, making them sensitive to tuning errors.

- The additional tuning parameters λ and μ allow fine-tuning of gain and phase margins, increasing the range of stable operations.

- The phase-lead effect of fractional differentiation improves the frequency response, ensuring better stability in systems with delays and nonlinearities.

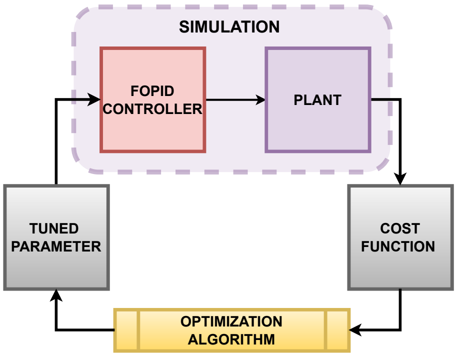

Fig. 4. Design and optimization of $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controllers [84]

Table 1. Comparison of FOPID tuning methods (Heuristic, Analytical, and Optimization-Based)

Tuning Method	Approach	Key Features	Advantages	Best Suited For
		Heuristic Methods		
Ziegler-Nichols (ZN) Extension	Classical heuristic	Step response-based tuning	Simple, widely used	Industrial PID tuning
Cohen-Coon Method	Empirical rule-based	Process reaction curve fitting	Good for open-loop processes	Chemical and thermal systems
Chien-Hrones-Reswick (CHR)	Time-domain heuristic	Overshoot and settling time-based	Improved disturbance rejection	Servo and regulation control
		Analytical Methods		
Pole Placement	Algebraic approach	Direct pole positioning	Precise stability margins	Strict stability requirement systems
Internal Model Control (IMC)	Model-based	Inherent robustness design	High stability, disturbance rejection	Process control systems
Frequency-Domain (FD) Methods	Frequency-based	Uses Bode/Nyquist plots	Ensures robust stability	Noisy environments
		Optimization-Based Methods		
Genetic Algorithm (GA)	Evolutionary optimization	Global search strategy	Avoids local minima	Nonlinear and uncertain systems
Particle Swarm Optimization (PSO)	Swarm intelligence	Fast, adaptive convergence	Computationally efficient	Online tuning applications
Grey Wolf Optimization (GWO)	Metaheuristic search	Balances exploration	Robust against local minima	Complex and uncertain systems
Ant Colony Optimization (ACO)	Bio-inspired search	Path-finding approach	Suitable for combinatorial tuning	Complex industrial processes
Simulated Annealing (SA)	Probabilistic search	Gradient-free method	Avoids poor local optima	Dynamic system optimization

3. Overview of Programmable Logic Controllers (PLCs)

Another such invention was PLC, whose basic structure is as shown in Fig. 5. which is closely tied to the evolution of industrial automation [35] and emerged as a solution to streamline manufacturing processes. Future developments in their smart versions, cybersecurity, and cloud integration will further expand their applications, making them a vital component of Industry 4.0. They are crucial for industries as they guarantee high-speed control of equipment and processes, minimizing human error and involvement.

They analyze the sensor input, modify settings, and manage actuators to allow equipment to make real-time decisions. Keeping in check energy consumption and managing energy [36] plays a critical role in controlling various industrial processes, which is generally thought of as a secondary consideration. Optimizing the energy performance of PLC-driven systems to minimize overall consumption and carbon emissions can be achieved by implementing energy-efficient control strategies—such as optimized start-stop operations, dynamic load management, and integrating advanced algorithms like fractional order controllers.

Their modern variant is armed with better data processing capabilities, integration features, and communication protocols that are well suited for Industry 4.0 [3], [37], [38]. Automating these processes with energy-conscious control strategies can lead to substantial energy savings.

Table 2. Comparison of FOPID tuning methods (Machine Learning, Intelligent Control, and Fractional-Based)

Tuning Method	Approach	Key Features	Advantages	Best Suited For
Machine Learning and Adaptive Methods				
Artificial Neural Networks (ANNs)	Self-learning	Data-driven tuning	Adaptive, handles nonlinearity	Time-varying, nonlinear systems
Reinforcement Learning (RL)	Learning-based	Continuous feedback learning	Self-optimizing control	Dynamic, uncertain environments
Deep Learning-Based Optimization	Neural-based search	Uses deep structures	High adaptability	Complex industrial automation
Fuzzy and Intelligent Control Methods				
Fuzzy Logic-Based Tuning	Rule-based adaptation	Human-like decision-making	Handles complex nonlinearity	Robotics and automation
Neuro-Fuzzy Tuning	Hybrid approach	Combines ANN and fuzzy logic	Self-learning	Intelligent control systems
Adaptive Neuro-Fuzzy Inference System (ANFIS)	Hybrid intelligent control	Data-driven inference	Self-tuning capabilities	Autonomous and smart systems
Fractional Approximation and Model-Based Methods				
Oustaloup's Recursive Approximation (ORA)	Model-based	Approximates fractional derivatives	Effective for digital implementation	Real-time embedded control
CRONE Control (First, Second, Third Gen.)	Frequency-based fractional design	Robust fractional controller tuning	High stability under uncertainty	Uncertain and variable conditions
Mittag-Leffler Function-Based Tuning	Analytical	Uses Mittag-Leffler functions	Better transient response	Systems with memory effects
Grünwald-Letnikov Approximation	Discrete numerical method	Fractional differentiation via sum approximation	Computationally efficient	Digital control implementation
Caputo Derivative-Based Tuning	Fractional calculus	Alternative to Grünwald-Letnikov	More flexible in boundary conditions	Long-term memory systems

There is potential for improving the estimation and control of energy systems so that errors are reduced as compared to the traditional methods using optimized control laws [26]. Choosing a

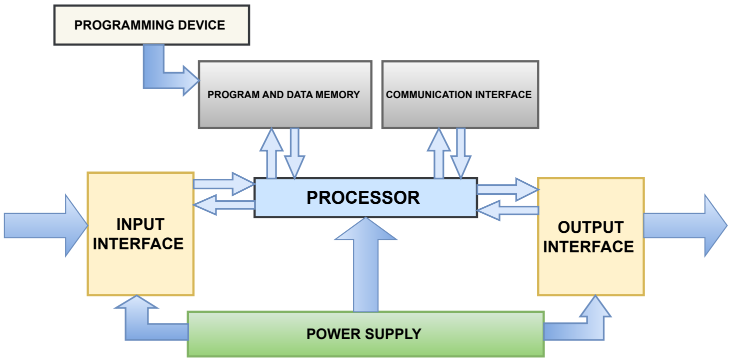

Fig. 5. Block Diagram Representation of Programmable Logic Control

finite sampling period (T_p) is very important as the process variable is discrete and the output is available at particular sample points. A smaller sampling period means higher accuracy [54]. Real-time computational capabilities allow for the control of the system in a closed-loop manner, with feedback from the sensor used to adjust the control signal. The tuning procedures are generally used to optimize the performance of the controller by selecting values that balance system response speed and stability, minimizing the control error while avoiding excessive oscillations or instability. The tasks are present in 'n' cyclic classes. Each cyclic class has a priority; cyclic class 2 will not start before cyclic class 1 is finished. The input reading is the A-to-D conversion, and the output writing that is the D-to-A conversion, as well as the time for task calculation, has to be considered. From Fig. 6, it is evident that the duration of task $\tau_{\text{calc}}(t) \ll T$, where T is the sampling period of a certain class underscores the selection of a proper approximation technique that will not occupy too much memory and consume processor time.

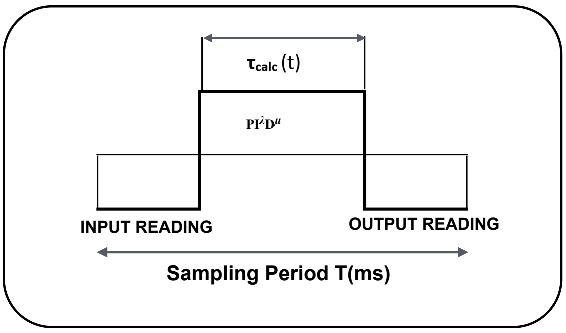

Fig. 6. Sampling period of PLC

PLCs use specialized programming languages defined by the IEC 61131-3 standard. Simulink PLC Coder is an add-on to Simulink that allows the generation of hardware-independent IEC 61131 compatible structured control language or ladder diagrams from Simulink models. The tool automatically generates IEC 61131-3 structured text (ST) code from Simulink models, enabling seamless deployment to PLCs and also supports popular PLC platforms like Siemens, Rockwell Automation (Allen-Bradley), Beckhoff, etc, as shown in Table 3.

The Simulink modules are generated as different documents which can then be deployed on different PLC software, such as the TIA Portal that particular systems use. It allows the generation of a Simulink model to IEC-61131 standard language for many integrated development environments. Code generation is the process of automatically converting a high-level model or algorithm into executable code for a target platform [82]. In the context of the Simulink PLC Coder, this means transforming Simulink/Stateflow models into IEC 61131-3 Structured Text (ST) that can be deployed in PLCs.

Table 3. Supported PLC platforms for simulink PLC coder

PLC Brand	Software	Supported PLCs
Siemens	TIA Portal	S7-300, S7-400, S7-1200, S7-1500
Rockwell	Studio 5000	ControlLogix, CompactLogix
Beckhoff	TwinCAT 3	CX-Series, IPC Controllers
Schneider	EcoStruxure	M241, M251, M262
B&R	Automation Studio	X20, X90 Series
ABB	Automation Builder	AC500 PLCs
Mitsubishi	GX Works3	iQ-R, iQ-F, FX5U
Omron	Sysmac Studio	NJ/NX Series
Bosch Rexroth	IndraWorks	IndraControl PLCs
Phoenix	PC Worx	Axiocontrol PLCs

The process ensures that control algorithms designed in Simulink can be efficiently implemented in industrial automation systems without the need for manual coding. The points to keep in mind are as follows:

- PLC Scan Cycle: The scan cycle involves reading inputs, executing logic, and updating the outputs. It runs on deterministic timings, so Simulink has to match this behavior during simulation. The two-step solvers are variable and fixed-step solvers that have to be set in the solver section and set to the required step solver.

- Setting the Target IDE for PLC Code Generation: Different PLCs use different programming environments (IDE). The code generated should be compatible with the target PLC IDE. Selection has to be done from the list of supported versions by MATLAB.

- Creating an Atomic Subsystem for Code Generation: The use of atomic subsystems ensures that one block executes as a separate block independent of the other model elements. This allows the simplification of the PLC code generation and avoids unwanted optimization and elimination. In contrast, the nonatomic subsystem executes at the same level as other blocks. Selecting the block parameters and enabling them to be treated as an atomic subsystem.

- Generating PLC Code from Simulink: After the above step, the generate PLC code option is available in Simulink which creates a report that is code generation report that details the generated PLC code, eliminated or virtual blocks, and data type conversions.

- Handling Data Type Conversions in PLC Code: Simulink uses int 8 and the TIA portal; for example, Siemens PLC uses SINT so this type of conversion has to be done, which will ensure compatibility and prevent execution errors in the environment.

- TIA Portal Configuration and Integration of External Source Files: Once the files are generated in the TIA portal, they can be accessed through the External Source Files section by manual addition. The Structured Control Language (SCL) file that has been generated must be imported into this section. The option “Generate blocks from source” converts it to a function and instructs the TIA portal to interpret the SCL source code and generate an equivalent function block within the programming environment. TIA Portal automatically integrates the newly created function block into the main program structure. The conversion process also generates the database block that contains converted data types that were originally defined in SIMULINK. The structuring of these database blocks will be done on the basis of the target system. This process ensures that model-based designs from Simulink translate into real-world industrial automation applications with minimal manual coding effort. Simulink does not support PLC-specific input/output addresses, so after exporting, these tags must be manually updated in the TIA Portal under the PLC tags section.

4. Literature Review

4.1. Fractional-Order Proportional Integral Derivative Controller

4.1.1. Definition and Significance

An FOPID controller generalizes the traditional PID controller by incorporating fractional powers in proportional, integral, and derivative terms. FOPID controllers enhance system performance by providing better tracking, robustness, and stability. They are particularly effective in controlling nonlinear, time-varying, and complex systems, offering improved flexibility and adaptability over traditional PID controllers. Jayaram S and Venkatesan (2024) [9], [75] proposed the Approximate Generalized Time moments (AGTM) optimization technique for tuning FOPID controllers in a nonlinear real-time process to achieve a response closely matching a reference model. Here, it was found that these controllers offer superior adaptability, stability, and disturbance rejection compared to IOPID controllers, reducing the need for retuning. Experimental results confirm the AGTM-based controller's effectiveness in maintaining stability and handling parameter variations. Basant Tomar et al. (2024) [8] compare the conventional PID and the fractional order one concluding that the latter is better in performance compared to the former. Using certain cost functions, they have used optimization algorithms and concluded the above.

Shah et al. (2024) [2] state that there is a lot of scope for fractional order control research related to patents, hardware implementation, non-integer control, anomalous diffusion, chaos, and task analysis.Shan et al. (2023) [10] show simulation results that compare other methods, such as traditional Internal Model Control (IMC) and Zeigler-Nichols(Z-N); the fractional-order IMC can effectively control the set value tracking and suppress the load disturbance during batch cooking. It provides better setpoint tracking dynamic performance and better robustness to system parameter perturbations than IMC and Z-N when the models do not match. In addition, the method is easy to implement and can be tuned online. Bartosz Puchalsk (2022) [11] highlighted the implementation of neural approximated operators on digital platforms operating in real-time conditions, which do not have significant computing power, for instance, PLC, Programmable Automation Controller (PAC), microcontrollers [71], or FPGA boards. Petras I. and Terpak J. (2019) [12] proposed a method based on fractional-order differentiation/integration using the Grünwald–Letnikov definition of the fractional-order operators. Observation of important properties in the time series and decisions in real process control can be made. Shantanu Das et al. (2018) [13] have designed the Fractional Calculus Engineering Laboratory, claiming that there are no commercial translations of such work and hope that these systems are used for energy efficiency, energy conversion, and enhanced robust systems. Fuel/energy efficient controls are realized via fractional calculus.

4.2. PLC-Code Generation

4.2.1. Definition and Significance

The process automates the creation of control programs for PLCs, improving efficiency, reducing errors, and ensuring standardization. It integrates with model-based design tools like MATLAB/Simulink, enabling streamlined deployment across different PLC hardware. This process enhances industrial automation, supports Industry 4.0, and smoothens debugging and maintenance. Banerjee A and Choppella V (2025) [14] stated that controller synthesis is pivotal in automating control system design from formal specifications and enhancing industrial system verification and optimization processes. This ensures system stability and reliability by automating processes. MAT-

LAB's simulation capabilities can be used to check if the system behavior adheres to the Linear Temporal Logic (LTL) specifications, enabling the identification of potential timing issues or design flaws early on. It is used in model checking, formal verification, and industrial automation to ensure that control systems behave as expected under different conditions.

Damian McCarthy, et al. (2022) [15] use MathWorks® Simulink® to model manufacturing equipment, enabling digital validation through simulations and claim that no other model-based platform has integrated PLC code generation capabilities, to the best of their knowledge, making Simulink uniquely suited to industries where equipment control is predominantly programmed with PLCs. Its wide range of compatible PLC software targets PLC software vendors, such as Rockwell, Siemens, B and R, PLC Open, and others, enabling the equipment designer to take a vendor-agnostic approach to equipment design. Sykora et al. 2021 [16] employed MATLAB to generate an implementation code, utilizing a code created in Simulink. Identifying the system through an industrial control setup, creating a control loop model, and generating a code implementable in the control setup.

4.3. PLC-Based Control Systems

4.3.1. Definition and Significance

A PLC-based control system is an industrial digital computing system that automates real-time process control by monitoring sensor inputs, executing logic-based algorithms, and generating actuator outputs for precise machinery and system operation. PLCs are highly durable and adaptable, making them ideal for industrial automation with real-time processing, seamless network integration, and reduced maintenance complexity. They enhance safety and efficiency through fail-safe mechanisms, diagnostics, and automated monitoring, ensuring optimal system performance. Basant Tomar et al. (2024) [8] have used the Wonderware Intouch SCADA software to visualize the entire plant's operations, and Sysmac Studio automation software is utilized to program the PLC using the ladder programming language along with algorithms such as GA, PSO, and ACO. Also, it is pointed out that to get the best performance GWO and AEO can be employed. Miguel Terrón-Hernández, et al. (2024) [17] developed a control algorithm for an electronic system using the PLC, which programmed the function for solar tracking. In MATLAB, data on power, angular velocity, and torque are incorporated to set control parameters for the solar tracking system. Faisal Abbas et al. (2024) [18] proposed that the FOMCON toolbox can be used in the Windows-based MATLAB application to program the suggested control system. The toolbox approximates fractional-order operators in the continuous domain using the Oustaloup recursive approximation. Mehta et al. (2024) [19] have done a comparative study concerning the developed hardware PLC-based test bench in view of the MATLAB simulation platform to validate the performance and control approach for stable microgrid (MG) operation.

Furthermore, using the proposed prototype, the decision-making algorithm will incorporate parameters such as system voltage, system current, overall power factor, and harmonics across the electrical load, etc. Duhe et al. (2023) [20] point out that memory requirements or memory effects are extensive, as fractional order control involves storing historical data or system memory effects. Wenjuan Shan et al. (2023) [47] used a MATLAB program to exchange data between OPC Toolbox and configuration software WinCC. WinCC received the output of MATLAB and the S7-300 PLC served as the basis for the controller. Gude et al.(2022) [21] presented a novel control hardware architecture for the practical implementation of integer- and fractional-order control algorithms on a real-time target. Jamil et al.(2022) [22] have concluded that the dragonfly algorithm used in bioreactors converges faster [74], which can also minimize the objective function. It is concluded that FOPID controllers [72] are more robust and efficient in the industrial sector for temperature control. Mozaryn et al. (2021) [23] implemented an FOPID controller on a PLC Siemens S7-1200 controller, and approximation was done using the CFE [67]. This solution was tested via simulations and experiments on the laboratory test stand and compared with a standard IOPID controller. Mystkowski et al. (2018) [24]worked on the MPS water Compact Workstation developed by Festo, the PLC Simatic controller developed by Siemens, and PC computer with TIA portal +WinCC and MATLAB/Simulink software. The sampling time is 0.15s. 93.5 percent fit suggests that the model's predictions closely match the observed data used during the training phase. The total time delay of the system is equal to 8.544s when added to the power start-up pump and the A/D terminal delay. Fractional orders, gain margin, and phase margin were used in simulations of Power Series Expansion (PSE) utilizing the Euler and Tustin methods. Gains were modified in accordance with the recalculated parameters.

4.4. Energy-Efficient Control Strategies

4.4.1. Definition and Significance

Through precise control, optimization of complex processes, and reduction of energy waste, FOPID controllers can play a crucial role in achieving energy-efficient industrial operations. Alilou et al. (2023) [26] conclude that fractional-order systems are widely applied in engineering, including in renewable energy and energy storage. Key challenges include integrating fractional techniques with intelligent estimation for uncertain dynamics, designing self-regulated systems to handle faults, and addressing controller delays in practical performance. Additionally, studying estimation errors, system uncertainties, and external perturbations is crucial for modeling, control, and stability. Tobajas et al. (2022) [25] have noted that energy consumption in industrial automation is a critical concern, with researchers focusing on intelligent optimization techniques to improve energy efficiency.P. Warrier and P. Shah (2021) [27] emphasize that fractional order controllers offer promising advancements for efficient, smart, and sustainable energy systems, essential for meeting Industry 4.0 standards. Tepljakov A et al. (2018) [28] state that the FOPID has proven to provide better control signals, but the dearth of research on energy-efficient algorithms for these intricate systems prevails. Ranganayakulu R et al. (2018) [29] have compared conventional PID and FOPID tuning methods. Metrics based on the Integral of Absolute Error (IAE), Total Variation (TV), and Maximum Sensitivity (M_s) were examined. First-order plus dead time processes were checked for robustness with increasing L/T ratio with respect to IAE, TV, and M_s, and an optimal tuning method was recommended. This type of research is limited and has to be explored more.

4.5. Literature Insights

From the literature review section, it was deduced that FOPID controllers offer greater flexibility and robustness compared to classical PID controllers. Their particular usefulness for handling systems with high uncertainty and nonlinear behavior gives them an edge over the other controllers, thereby enabling them to handle nonlinear processes in nature. The thorough literature review shows that fractional order PIDs are a very lucrative choice for industrial automation that needs to control such natural processes. The implementation of these controllers on the PLCs is very sparsely explored, which is aimed to be explored through this review. To optimise FOPID control in industrial PLC applications, machine learning techniques, such as reinforcement learning and nature-inspired algorithms, need to be further explored. Automated code generation reduces manual implementation effort, enhances computational efficiency for real-time FOPID execution, optimizes control performance, minimizes energy consumption, and ensures seamless cross-platform compatibility for industrial PLCs.

5. Review Methodology

5.1. Implementing Fractional Order Controllers on PLCs

This review synthesizes recent research efforts, emerging methodologies, and practical considerations surrounding the implementation of energy-efficient Fractional-Order PID (FOPID) controllers on Programmable Logic Controllers (PLCs). The analysis is structured across six critical phases, reflecting the design-to-deployment lifecycle of such systems. The reviewed studies highlight both theoretical advancements and industrial applicability, with a focus on control performance, computational efficiency, and energy sustainability.

- System Analysis and Problem Formulation: Research consistently highlights limitations in scan cycle durations, memory allocation, and processor speed as major barriers to implementing computationally intensive algorithms like fractional-order controllers. Comparative energy profiling of conventional PID and FOPID controllers, utilizing tools like power analyzers, current sensors, and PLC diagnostics to assess energy draw during closed-loop control.
- FOPID Optimization for PLC Deployment: The literature reveals a growing focus on low-complexity implementations of fractional operators for PLC environments. ORA remains the most frequently adopted method due to its favorable frequency-domain characteristics and amenability to precomputed coefficient use .Use of recursive structures and lookup tables to eliminate the need for real-time complex number operations or high-order filters.
- Energy-Efficient PLC Execution Strategies: A substantial body of work explores real-time optimizations at the PLC programming level. These efforts aim to reduce processor load, shorten execution cycles, and minimize unnecessary computations. Task prioritization and interrupt-driven control logic allow critical tasks to preempt less essential routines. Optimization of Structured Text (ST), Function Block Diagram (FBD), and Ladder Logic (LD) to reduce code complexity.Adaptive sampling and event-triggered control, which align execution frequency with system dynamics to reduce computational waste.Low-power operation modes and multi-core processing, where supported, to balance performance with energy usage.
- Simulation and Evaluation Approaches: Simulation plays a pivotal role in evaluating controller performance prior to deployment. The majority of studies leverage MATLAB/Simulink environments, often augmented with toolboxes like FOMCON for modeling and analyzing fractional-order systems. The performance metrics are IAE, ISE, ITAE, and ISCS (Integral of Squared Control Signal) for control and energy evaluation. Additional metrics include computational load indicators, such as CPU usage and memory allocation, and control signal smoothness and actuator wear estimations. Simulation results often serve as precursors to Hardware-in-the-Loop (HIL) testing and deployment on PLC platforms.
- Experimental Validation on Industrial PLCs: Experimental efforts in recent literature have expanded beyond simulation, with deployment on Siemens, Allen-Bradley, and Beckhoff PLCs. HIL testing frameworks allow integration with real-time plant emulators and enable fine-grained assessment of controller performance under actual PLC constraints. Experimental observations include real-time measurements of latency, scan times, and processor utilization .Energy consumption tracking and evaluation of control performance under step input and disturbance scenarios. These experiments validate the theoretical feasibility and energy-saving potential of optimized FOPID implementations.
- Optimization Strategies and Industrial Feasibility: Application of multi-objective optimization frameworks to identify trade-offs between energy use and transient response. The use of adaptive FOPID controllers, capable of real-time retuning based on process dynamics and performance feedback.Development of modular, scalable design templates to enable cross-platform deployment with minimal reconfiguration.

Further expanding this is a detailed technical outline that can be followed to achieve the above points. The two subsequent steps are first to design and tune the fractional order PID [83] and then implement it on the PLC. The system design will use the concepts of fractional calculus to develop a fractional order controller algorithm that will be put into practice using the PLC programming environment. Simulation and modeling are done using tools like MATLAB/Simulink to test FOPIDs behavior in comparison to traditional controllers. The model focuses on energy consumption, system response time, etc. Data on energy consumption, system performance (settling time, overshoot, etc.), and stability can be collected. Statistical analysis is a good and popular approach to conduct a comparison of the energy efficiency and performance of the FOPID versus traditional controllers [56]. For the system design and modeling, the plant model designing is the first step. Any control system is dynamic and has nonlinear characteristics. The primary goal of such a system is to maintain a setpoint despite disturbances or changes in the environment. For representing such systems, transfer functions are used, which mathematically represent the relationship between inputs and outputs in the frequency domain. The transfer function will help to tune the controllers. The entire system's transfer function includes the process, actuator, and sensing dynamics:

$$G_{\text{overall}}(s) = G_v(s)G_P(s)G_H(s) \tag{5}$$

For example, if the process is approximated using the FOPTD(fractional order plus time delay) model, the transfer function is as follows:

$$G_f(s) = \frac{K_f e^{-t_d s}}{\tau s + 1} \tag{6}$$

This will help identify system dynamics like step response and estimation of errors. Before implementation, the system's response to control inputs or disturbances can be predicted. This can be done using the MATLAB system identification toolbox. As shown in Equation 1, the fractional-order, the integrodifferential operator can be described by different definitions, which are GL, RL, and C definitions. The digital modeling of the fractional order operator is done using any one of these definitions. The implementation of the operator in Equation on a digital platform requires the application of the discrete approximant. Tuning a FOPID controller involves finding the optimal set of parameters that define the controller's behavior, which is done using the MATLABs FOMCON toolbox [65]. The most known methods are the PSE [68] and CFE. The accuracy of both of these types of approximations is estimated by the calculation of the cost functions and then the reduction of these that are the Integral of Squared Error (ISE), Integral of Absolute Error (IAE) and Integral of Time-weighted Absolute Error (ITAE) as shown in the equations (7), (8), (9).

$$ISE = \int_0^\infty e^2(t)\, dt \tag{7}$$

$$IAE = \int_0^\infty |e(t)|\, dt \tag{8}$$

$$ITAE = \int_0^\infty t|e(t)|\, dt \tag{9}$$

The implementation of an elementary fractional-order plant, represented by a transfer function in the s-domain, on a PLC platform utilizing normalized software tools is based upon the approximation order of the model, with higher-order approximations delivering superior accuracy. The adoption of a particular approximant that helps in the realization of the fractional order element depends upon optimal order selection, execution time, and memory [73]. There has to be a tradeoff between accuracy and memory capacity. The input and output signals out of a FOPID are normally analog. The current outputs are 0-20 mA or 4-20 mA. In the case of voltage outputs on the module, it is 0-10 V, ± 5 V, or 0-5 V. The actuators accept either 0 or 1 as input. For simulation and parameter tuning, the control algorithm is implemented in the MATLAB/Simulink environment. The functional blocks are created.

Direct implementation is not possible in Simulink. Hence, depending on the approximation method selected, the $s^{\lambda}$and s^{μ} are implemented in a block that is dedicated to the approximation method. This MATLAB function block can be executed directly in Simulink. In MATLAB, the object-oriented programming technique establishment of the FOTF-class, with which the fractional-order transfer function can be expressed, is present. It is used to describe FOTF block interconnections. For instance, for the FOTF class, a folder named @fotf should be created first [22]. This can be done in MATLAB/Simulink [20], which can be used to realize the fractional order operators. Approximation helps in reducing computational complexity as in converting higher order systems to lower order systems, simplifies nonlinear models, and also helps in improving real-time implementation.

This process leads to a loss of accuracy in system behavior and deviation from desired performance. More fine-tuning methods can be done using optimization methods to refine the parameters, stability under different operating conditions, and energy-aware tuning that helps minimize power consumption. By fine-tuning system parameters, optimization ensures that approximated models work as close as possible to their ideal counterparts while being computationally efficient. Fig. 7. shows the proposed way in which the FOPID is realized on PLC.

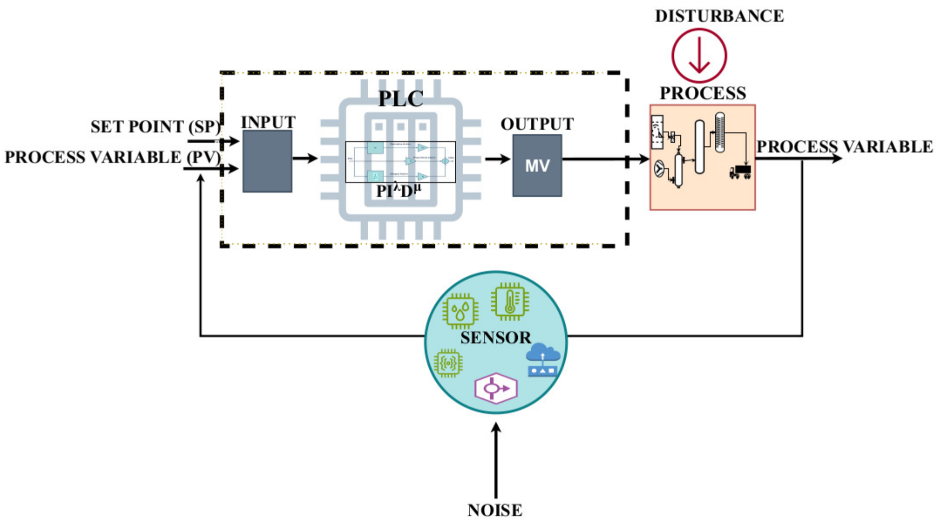

Fig. 7. Design and deployment of FOPID in industrial PLC systems

The elements are input devices or sensors that detect physical changes like temperature, pressure, and proximity and send the signals to the PLC. The input module receives the signals from the sensors and converts them to signals that PLC can understand. The brain is the central processing unit (CPU) where the input data is processed and executed, after which appropriate output signals are sent along with which to and from communication with the other devices is done.

The output module then controls external devices like actuators, etc. The FOPID sets the desired setpoint and the difference between the desired setpoint and the actual process output. The process plant is the actual plant that is being controlled. The feedback loop is created where the output is fed back and compared with the error signal. In essence, while the PLC block diagram shows the interaction between hardware elements in an automation process, the FOPID block diagram focuses on control system elements for precise and flexible process control using fractional calculus. Combining the above two, that is implementation of FOPID controllers in PLCs represents a significant advancement in industrial automation.

The PLC is the hardware used to implement the FOPID control strategy. Existing control systems that have the PLCs with FOPIDs integrated are a promising avenue in research and development; for example, some experimental setups use ethernet communication between the PLCs having the fractional order control algorithm. The hardware in loop simulation involves TCP/IP communication between Simulink and the PLC [39]. It is a well-known fact that work/effort towards reducing or better controlling processes to save energy exists. Major efforts to implement fractional order theory to control systems have been made. Leveraging the principles of fractional calculus, the ability to reduce energy consumption and improve operational efficiency is achievable but less explored and quantified. Smart manufacturing objectives can be achieved by the use of energy-efficient control strategies that use fractional order controllers in PLC. The aim is to explore the application of FOPID to develop energy-efficient control strategies and compare it with conventional PIDs. The implementation of FOPID controllers in PLCs has gained traction because of their control capabilities and robust control action as compared to IOPID controllers [40].FOPID controllers exhibit improved robustness against disturbances and uncertainties.

All the past outputs have to be memorized due to memory effects in FOPIDs hence the discretization and approximation methods are used common one being the GL operator of order $\alpha \in (n-1, n)$ where n is the step size and could be any real number [24]. These approximation methods simplify the implementation of fractional-order controllers, making them more feasible for industrial use by approximating the fractional derivatives and integrals to integer-order equivalents. Tuning the parameters of the FOPID controller involves determining the appropriate values for the proportional, integral, and derivative terms, as well as the fractional orders of each term. For the translation of fractional order operations to discrete-time algorithms suitable for PLCs, these methods are crucial. FOPIDs have been successfully implemented on PLCs in industrial scenarios such as temperature control [24] by implementing in pipelines to maintain desired temperature, hydraulic systems [41] for providing enhanced precision and response times, and water level control [8], [23] but have not focused on energy efficiency. The different tuning methods for the FOPIDs as discussed in [42], [43] are optimization with integral criteria, autotuning, and so on. The structure is implemented in the PLC controller with a high-level Structured Text (ST) language, for instance with Simulink PLC Coder toolbox [44]. Utilizing the PLC and SCADA systems, a framework is developed for supervising and, in turn, regulating the temperature and data collection [8], [12]. The factors influencing the method selection are PLC processing power due to computational complexity, frequency range of interest, and the need for more resources due to complex models. Also, for a FOPID controller, approximations used are complicated and need more resources for computation [28]. Tracking systems consume energy to operate motors or actuators. This self-consumption must be factored into the overall system efficiency. FOPID controllers provide fine-tuned control over the tracking system's motors or actuators by managing their movements more efficiently.

Implementation of PLC can be done with the help of software tools and object-oriented programming using the IEC 61131 standard [50], [51]. For making the system behave as desired, the amount of energy that is the control energy used by the controller is known as control effort [52]. The control effort refers to the energy or magnitude of the input signal u(t) that the controller uses to achieve its goal, for example, tracking references and minimizing errors. Mathematically, it is represented as seen in Equation (10) [53].

$$E = \int_0^T u(t)^2 \, dt \tag{10}$$

This quantifies the total work done by the controller. There are two types of efforts that are cheap control effort and fixed control effort in which there is no penalty for using more energy (control input); in contrast, a penalty is placed if the controller exceeds a pre-defined threshold respectively [55]. The reason to fix the control effort is that actuators (e.g., motors or valves) have finite capacities and can't produce infinitely high or sustained control inputs. To operate for long hours, conservation of energy is needed. Control signals beyond a certain level can cause saturation, instability, and damage. A constraint is fixed in controller design now, the performance metrics must be balanced with this constraint. In an optimal control problem, the cost function might now include a term that penalizes excessive control input. If the control effort is "cheap", the controller might overuse a heating/cooling element to maintain precise temperature, regardless of power consumption. If the control effort is "fixed", the system limits how much energy the controller can use.

This may result in slower temperature adjustments or slight deviations from the setpoint, balancing precision with energy efficiency. Reduced energy might weaken the controllers' ability to keep the system stable, especially under challenging conditions like disturbances. Hence, a trade-off between energy efficiency, error, and stability has to be made. The parameters associated with FOPID are Kp, Ki, Kd, λ, and μ, which pose challenges for implementation due to their high dimensional nature [50]. Hence, the limited computational resources in PLCs make the process challenging. Since there is a large search space, exploring all the possible combinations is an arduous task and increases exponentially. Also, the complexity associated with the PLCs being resource-constrained leads to a lack of processing power to handle such conditions efficiently. Timing constraints within industrial settings also place a burden on optimization algorithms that require multiple iterations that exceed the allowable computation time of PLCs and convergence exceeding practical limits. Algorithms get trapped in local minima as there are multiple present in the search space leading to suboptimal parameter values.

A lack of deterministic behavior due to a poor initial guess of FOPID parameters again leads to suboptimal values and convergence times. PLCs being designed for deterministic behavior /control overwhelms PLC processors. The objective function for FOPID tuning often involves minimizing an error metric. Hence, lightweight optimization techniques tailored for PLCs, leveraging parallel computation, model-free tuning, and combining global search methods with local optimization techniques for faster convergence highlight a deficiency in research. Small overshoots and faster convergence are observed in FOPID [66].

5.2. Hardware and Software Constraints in PLC-Based FOPID Implementation

In PLCs with limited resources, there are issues with processor power, memory constraints, and execution speed. A few challenges and the potential solutions are presented that have to be kept in mind while designing. Discrete-time approximations should be adjusted for a balance between accuracy and real-time feasibility because direct implementation of fractional calculus can be computationally costly. Computational overhead can be decreased by using strategies like reduced-order approximations and precomputed lookup tables. A PLC uses discrete scan cycles, and delays or skipped cycles may result if the control algorithm takes longer than the scan time. Due to their set execution times, PLCs make high-order computations more difficult than microcontrollers or industrial PCs. The sample time should be selected so that control calculations are finished in a single scan cycle in order to guarantee real-time execution; ideally, the PLC cycle time should be less than 70 % of the sampling time. The efficiency of execution can be further increased by using simpler numerical approximations. To optimize memory usage, fixed-point arithmetic can be used instead of floating-point calculations. Additionally, parameter reduction techniques can help minimize storage requirements for fractional differentiation operators, ensuring efficient implementation.To balance accuracy and execution speed, lower-order approximations should be used while maintaining effective performance. Additionally, adaptive control techniques can dynamically adjust the computational load based on system conditions, ensuring efficient real-time operation. While existing research mentions AI-based tuning, it lacks real-world case studies demonstrating feasibility in PLC-based execution. To address this, AI algorithms can be offloaded to edge computing devices or cloud-based optimization frameworks while the PLC operates with the optimized controller parameters in real time. Alternatively, lightweight heuristic methods can be integrated directly into PLC firmware for real-time parameter tuning.

When implementing a FOPID controller on a PLC with a focus on energy efficiency, the algorithm must balance control performance and computational efficiency. Key considerations include optimizing proportional (K_p), integral (K_i, λ), and derivative (K_d, μ) gains to minimize energy-intensive control actions while maintaining accuracy and stability. Efficient discretization of fractional operators, such as through Oustaloup or Grünwald-Letnikov approximations, is essential to reduce computational overhead. Sampling time must be carefully selected to balance precision and PLC energy consumption.The algorithm should incorporate energy-aware tuning methods, like online adaptive approaches or multi-objective optimization (e.g., genetic algorithms or particle swarm optimization), to dynamically adjust to changing system conditions [78]–[80].The role of AI-driven methods, such as machine learning (ML) and heuristic optimization, is gaining attention for FOPID tuning, but real-world adoption remains limited due to computational constraints. A review of autotuning methods for FOPID highlights that while techniques like reinforcement learning (RL) and deep learning (DL) can enhance controller adaptation, their feasibility on standard industrial PLCs is still an open challenge. Industrial applications typically favor simpler heuristic methods, such as GA or PSO, due to their lower computational demand compared to ML-based approaches. [80].

Another key finding is that industrial FOPID implementations must account for discrete-time effects, as improper tuning can lead to instability due to numerical inaccuracies in PLCs. Some recent studies suggest that hybrid approaches—where an initial FOPID design is refined using online learning techniques—can improve performance while keeping computational complexity manageable. Additionally, fault detection, predictive features, and efficient numerical solvers enhance system stability and reduce energy waste while minimizing execution time and memory usage, ensuring real-time responsiveness. Ultimately, combining these considerations enables an energy-efficient and reliable FOPID implementation on PLC. There is a lack of dedicated libraries and standardized tuning methods. Furthermore, suboptimal energy may arise from current control strategies in PLC systems when processes involve parameters such as loads and high inertia. An optimized framework is needed so that specific targets of energy consumption reduction can be met. This allows better system performance and also a future extension to leverage machine learning algorithms as an extension to the above.

The implementation of the FOPID controller on PLC is both challenging and less explored in industrial applications, unlike traditional PID controllers, which are well-supported by standard PLC hardware and software.The practical implementation of energy-efficient FOPID controllers on PLCs remains a challenge due to computational limitations, discretization inefficiencies, and trade-offs in control performance. While heuristic optimization methods offer a promising path, their industrial adoption depends on balancing real-time constraints with control accuracy. Future research should focus on developing lightweight AI-driven tuning methods and adaptive discretization techniques that align with PLC constraints while retaining the benefits of fractional-order control.FOPID controllers require numerical approximations for implementation on PLCs, given the non-integer differentiation involved. The Grünwald–Letnikov (GL) discretization method is commonly used due to its suitability for real-time execution on embedded hardware like PLCs. However, computational efficiency is a challenge. By following the flowchart as shown in Fig. 8, you can systematically achieve the desired goal.

6. Comparative Analysis and Discussion

- Computational Efficiency: Fractional order controllers show better control performance with a possible disadvantage, which is the complexity to implement on PLCs.Execution time is greatly decreased by using fixed-point approximations and recursive ORA implementations. For mid-range PLCs, near real-time control without compromising precision is possible with lookup-based techniques and precomputed fractional coefficients.
- Energy Savings Potential: An increasingly important component of sustainable automation is energy-

Vol. 5, No. 2, 2025, pp. 1211-1237

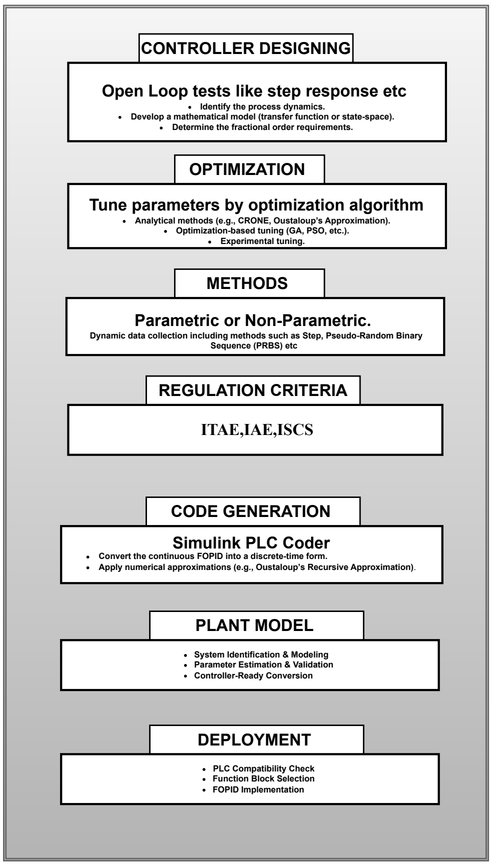

Fig. 8. Process flow diagram

aware control. Optimized FOPID designs reduce control-loop energy consumption by 25–35%. Energy savings are further enhanced by adaptive and event-driven sampling, particularly in slow-dynamics systems like thermal processes.

- Performance vs. Efficiency Trade-Offs: A recurring theme is the balance between control accuracy and computational resource use. While high-order approximations yield better performance, they also increase execution cost.

- Computational Intelligence: AI-driven tuning and adaptation are shifting FOPID design from static to dynamic paradigms.

- Industrial Viability: Standardization of PLC programming suites' fractional-order function components is needed. Fixed-point fractional operations are better supported in IEC 61131-3 environments. Industry-specific standards to facilitate incorporation into current control systems

Table 4 presents a comparative summary of recent studies implementing PID and FOPID controllers on industrial PLC platforms. Across varied applications—including temperature, power, and level control—FOPID controllers consistently demonstrate improved tracking, reduced rise and settling times, and enhanced signal quality. These results highlight the potential of FOPID control strategies to balance performance and energy efficiency in real-world PLC-based automation systems.

7. Conclusion

7.1. Challenges

While FOPID controllers offer superior performance in terms of robustness and adaptability, their practical implementation in PLCs remains constrained by computational complexity especially those relying on approximation techniques (e.g., Oustaloup, Grunwald–Letnikov, Tustin CFE) impose a significant computational burden on PLCs, particularly those with limited processing power, memory limitations, and the absence of standardized frameworks. Achieving real-time execution of fractional calculus operations on industrial PLCs remains a major bottleneck, often requiring trade-offs between accuracy and response time.

Table 4. Comparative summary of PLC-based control systems using PID and FOPID controllers

PLC Model	Controlled Process	Performance Metrics	PID vs FOPID
Omron NX NX1P2 9024DT1 [8]	Heat exchanger temperature control	PID (non-optimized): Rise Time = 25.02 s, Settling Time = 276.98 s, Overshoot = 19.48% FOPID (optimized using ACO): Rise Time = 12.97 s, Settling Time = 238.71 s, Overshoot = 19.2% Energy efficiency discussed qualitatively.	Optimized FOPID achieves significantly faster rise time and reduced settling time with comparable overshoot, contributing to more energy-efficient operation.
Siemens S7 1200 [23]	Pipeline temperature control	FOPID controller yields approximately 2× lower maximum error than PID, but has higher overshoot and longer settling time.	FOPID reduces maximum error substantially, while classical PID shows better overshoot and settling performance.
Manufacturer-supplied PLC (model unspecified) [44]	PWR thermal power control	PID shows excellent tracking but generates noisy control signals. FOPID with anti-windup achieves similar tracking with lower noise. No numerical values provided for overshoot or settling time.	PID achieves superior tracking accuracy; FOPID delivers cleaner control signals at the cost of increased computational complexity.
Festo MPS PA Compact Workstation [81]	Water level control	Optimized FOPID offers low overshoot, zero steady-state error, and short settling time. Energy efficiency is qualitatively validated.	FOPID outperforms classical PID in terms of overshoot, steady-state error, and energy efficiency, supported by experimental results.

Addressing these challenges is crucial for enabling widespread industrial adoption and enhancing energy efficiency in automated systems. One of the primary obstacles is the high computational load associated with fractional-order calculations, which poses significant challenges for real-time execution on resource-constrained PLCs. When compared to advanced control platforms like digital signal processors (DSP), field-programmable gate array (FPGA), and microcontrollers, PLCs have less processing power as they are built for discrete control and integer order computations compared to standard PID controllers. The limited processing power, sampling period and memory of PLCs can pose constraints when implementing the iterative and recursive algorithms required for fractional order calculations; hence, equations must be suitably discretized for implementation. Depending on the kind of process being monitored, this could result in an overestimation of the controller's performance under various circumstances. During operation, noise, disturbances, and changes in process dynamics frequently limit precision, such as the controller gain. Another critical challenge is the lack of universal standards and software toolchains for integrating FOPID into PLC programming environments. Unlike conventional PID controllers, which are well-supported in industrial automation software, FOPID controllers require specialized libraries and parameter-tuning strategies. Potential solutions include optimized numerical approximation techniques, lookup table-based implementations, and hardware acceleration (e.g., FPGA or GPU-assisted computation). While metaheuristic algorithms have shown promise in optimizing FOPID parameters, they are computationally intensive and may not be practical for online adaptation in low-power PLCs. The development of standardized function blocks, energy-aware tuning algorithms, and adaptive control strategies will play a key role in making FOPID a viable option for real-time industrial applications. Although many studies claim energy-efficient operation, few provide standardized metrics or benchmarks to quantify energy savings, leading to inconsistent evaluations across the literature.

7.2. Future Research Directions

A noticeable gap exists between theoretical advancements and the robust, practical tools industries require. Current research shows a lack of cost-effective, energy-efficient solutions. While significant progress has been made in the field of energy-efficient FOPID control for industrial automation, several technical and practical challenges remain. Addressing these issues through computational optimizations, standardization efforts, and interdisciplinary research will be critical for unlocking the full potential of FOPID in next-generation PLC-based automation systems. There is a need to emphasize the need for interdisciplinary collaboration between fractional calculus experts and PLC programmers that can take the best of both worlds and use it to the benefit of humankind. Future advancements in hardware, software integration, and energy-aware tuning methodologies will determine the feasibility of deploying FOPID controllers at scale in industrial environments. Real-time hardware solutions and software-based simulations of fractional controllers are limited. Extensive studies comparing the energy usage of FOPID and IOPID in PLC applications is the need. An opportunity is present to explore how FOPID controllers can be enhanced through Industry 4.0 technologies to enable more adaptive, efficient, and predictive control strategies .Development of Lightweight Approximation Techniques tailored for resource-constrained PLCs.Integrating Industry 4.0 technologies, such as machine learning and IoT, can enable the adaptive tuning of FOPID parameters in real time, enhancing system resilience and energy efficiency. Creating modular and hardware-agnostic FOPID libraries compatible with major PLC brands would support wider industrial adoption. Future tuning methods should incorporate energy consumption directly into their objective functions to optimize both control performance and power efficiency. There is a strong need for more hardware-in-the-loop (HIL) and real-world experiments to evaluate FOPID controllers across varied industrial scenarios, including non-linear and time-delay systems. In a nutshell, the paper highlights the recent advancements, challenges, and future prospects of energy-efficient FOPID control in industrial PLC-based automation. There is a need for automated verification to enhance safety and reliability, along with better integration with Industry 4.0 and IoT. Research on AI/ML-driven PLC programming remains limited, offering the potential for automation and intelligent code generation.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: No funding was provided.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] I. B. Campos *et al*., “Energy efficiency assessment: Process modelling and waste heat recovery analysis,” *Energy Conversion and Management*, vol. 196, pp. 1180-1192, 2019, https://doi.org/10.1016/j.enconman.2019.06.074.

[2] P. Shah and S. Agashe, “Review of fractional PID controller,” *Mechatronics*, vol. 38, pp. 29-41, 2016, https://doi.org/10.1016/j.mechatronics.2016.06.005.

[3] Y. Chen, I. Petras and D. Xue, “Fractional order control - A tutorial,” *2009 American Control Conference*, pp. 1397-1411, 2009, https://doi.org/10.1109/ACC.2009.5160719.

[4] R. El-Khazali, “Fractional-order $PI^{\lambda}D^{\mu}$ controller design,” *Computers and Mathematics with Applications*, vol. 66, no. 5, pp. 639–646, 2013, https://doi.org/10.1016/j.camwa.2013.02.015.

[5] I. Petráš, “Tuning and implementation methods for fractional-order controllers,” *Fractional Calculus and Applied Analysis*, vol. 15, pp. 282–303, 2012, https://doi.org/10.2478/s13540-012-0021-4.

[6] J. Iqbal *et al.*, “Nonlinear control systems- A brief overview of historical and recent advances,” *Nonlinear Engineering*, vol. 6, no. 4, pp. 301–312, 2017, https://doi.org/10.1515/nleng-2016-0077.

[7] I. Podlubny, “Fractional-order systems and PI/sup /spl lambda//D/sup /spl mu//-controllers,” in *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 208-214, 1999, https://doi.org/10.1109/9.739144.

[8] B. Tomar, N. Kumar, and M. Sreejeth, “PLC and SCADA based temperature control of heat exchanger system through fractional order PID controller using metaheuristic optimization techniques,” *Heat and Mass Transfer*, vol. 60, pp. 1585–1602, 2024, https://doi.org/10.1007/s00231-024-03509-5.

[9] S. Jayaram, and N. Venkatesan, “Design and implementation of the fractional-order controllers for a real-time nonlinear process using the AGTM optimization technique,” *Scientific Reports*, vol. 14, 2024, https://doi.org/10.1038/s41598-024-82258-1.

[10] W. Shan, Y. Wang, and W. Tang, “Fractional order internal model pid control for pulp batch cooking process,” *Journal of Chemical Engineering of Japan*, vol. 56, no. 1, 2023, https://doi.org/10.1080/00219592.2023.2201288.

[11] B. Puchalski, “Neural Approximators for Variable-Order Fractional Calculus Operators (VO-FC),” in *IEEE Access*, vol. 10, pp. 7989-8004, 2022, https://doi.org/10.1109/ACCESS.2022.3143893.

[12] I. Petráš and J. Terpák, “Fractional calculus as a simple tool for modeling and analysis of long memory process in industry,” *Mathematics*, vol. 7, no. 6, p. 511, 2019, https://doi.org/10.3390/math7060511.

[13] S. Das, A. MV, J. AS and K. SW, “Energy/fuel efficient and enhanced robust systems demonstrated with developed fractional order PID controller,” *Innovative Energy and Research*, vol. 7, no. 1, p. 182, 2018, https://doi.org/10.4172/2576-1463.1000182.

[14] A. Banerjee and V. Choppella, “Challenges and opportunities in the industrial usage controller synthesis tools: A review of LTL-based opensource tools for automated control design,” *Results in Control and Optimization*, vol. 18, p. 100511, 2015, https://doi.org/10.1016/j.rico.2024.100511.

[15] D. McCarthy *et al.*, “A model-based approach to automated validation and generation of PLC code for manufacturing equipment in regulated environments,” *Applied Sciences*, vol. 12, no. 15, p. 7506, 2022, https://doi.org/10.3390/app12157506.

[16] T. Sýkora, M. Husák, O. Baštán and T. Benešl, “Automatic generation of a PLC controller based on a control system-identified model,” *Journal of Electrical Engineering*, vol. 72, no. 2, pp. 78–88, 2021, https://doi.org/10.2478/jee-2021-0011.

[17] M. Terrón-Hernández *et al.*, “Solar Tracking Systems in Compound Parabolic Concentrators,” *New Perspectives in Software Engineering*, vol. 1135, pp. 401–410, 2024, https://doi.org/10.1007/978-3-031-50590-4_25.

[18] F. Abbas, O. Saleem, L. Wang, and Y. Yan, “A self-adaptive fractional-order PID controller for the particle velocity regulation in a pneumatic conveying system,” *Transactions of the Institute of Measurement and Control*, 2024, https://doi.org/10.1177/01423312241277759.

[19] S. Mehta, and P. Basak, “Development of PLC-based hardware test-bench prototype for solar-wind-battery-based microgrid system’s control algorithm validation,” *Electric Power Components and Systems*, pp. 1–20, 2024, https://doi.org/10.1080/15325008.2024.2329326.

[20] K. E. Lucas-Marcillo *et al.*, “Novel Robust Methodology for Controller Design Aiming to Ensure DC Microgrid Stability Under CPL Power Variation,” in *IEEE Access*, vol. 7, pp. 64206-64222, 2019, https://doi.org/10.1109/ACCESS.2019.2915027.

[21] J. J. Gude and P. G. Bringas, “A novel control hardware architecture for implementation of fractional-order identification and control algorithms applied to a temperature prototype,” *Mathematics*, vol. 11, no. 1, p. 143, 2022, https://doi.org/10.3390/math11010143.

[22] A. A. Jamil *et al*., "Fractional-order PID controllers for temperature control: A review," *Energies*, vol. 15, no. 10, p. 3800, 2022, https://doi.org/10.3390/en15103800.

[23] J. Możaryn, J. Petryszyn and S. Ozana, "PLC based fractional-order PID temperature control in pipeline: design procedure and experimental evaluation," *Meccanica*, vol. 56, no. 4, pp. 855–871, 2021, https://doi.org/10.1007/s11012-020-01215-0.

[24] A. Mystkowski and A. Zolotas, "PLC-based discrete fractional-order control design for an industrial-oriented water tank volume system with input delay," *Fractional Calculus and Applied Analysis*, vol. 21, pp. 1005–1026, 2018, https://doi.org/10.1515/fca-2018-0055.

[25] J. Tobajas *et al*., "Resilience-oriented schedule of microgrids with hybrid energy storage system using model predictive control," *Applied Energy*, vol. 306, p. 118092, 2022, https://doi.org/10.1016/j.apenergy.2021.118092.

[26] M. Alilou *et al*., "Fractional-order control techniques for renewable energy and energy-storage-integrated power systems: A review," *Fractal and Fractional*, vol. 7, no. 5, p. 391, 2023, https://doi.org/10.3390/fractalfract7050391.

[27] P. Warrier and P. Shah, "Fractional Order Control of Power Electronic Converters in Industrial Drives and Renewable Energy Systems: A Review," in *IEEE Access*, vol. 9, pp. 58982-59009, 2021, https://doi.org/10.1109/ACCESS.2021.3073033.

[28] A. Tepljakov *et al*., "FOPID controllers and their industrial applications: A survey of recent results," *IFAC-PapersOnLine*, vol. 5, no. 4, pp. 25–30, 2018, https://doi.org/10.1016/j.ifacol.2018.06.014.

[29] R. Ranganayakulu *et al*., "A comparative study of fractional order $PI^{\lambda}/PI^{\lambda}D^{\mu}$ tuning rules for stable first order plus time delay processes," *Resource-Efficient Technologies*, vol. 2, pp. S136–S152, 2016, https://doi.org/10.1016/j.reffit.2016.11.009.

[30] J. D. Rojas *et al*., *Industrial PID Controller Tuning*, Spring Cham, 2021, https://doi.org/10.1007/978-3-030-72311-8.

[31] R. P. Borase *et al*., "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, pp. 818–827, 2021, https://doi.org/10.1007/s40435-020-00665-4.

[32] S. Das, *Functional fractional calculus*, Spring Berlin, Heidelberg, 2014, https://doi.org/10.1007/978-3-642-20545-3.

[33] J. C. C. Basilio, T. R. Oliveira, J. G. T. Ribeiro and A. Cunha, "Evaluation of Fractional-Order Sliding Mode Control Applied to an Energy Harvesting System," *2022 16th International Workshop on Variable Structure Systems (VSS)*, pp. 243-248, 2022, https://doi.org/10.1109/VSS57184.2022.9902105.

[34] P. Roy, B. Kar and B. K. Roy, "Fractional order PI-PD control of liquid level in coupled two tank system and its experimental validation," *Asian Journal of Control*, vol. 19, no. 5, pp. 1699-1709, 2017, https://doi.org/10.1002/asjc.1487.

[35] E. R. Alphonsus and M. O. Abdullah, "A review on the applications of programmable logic controllers (PLCs)," *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 1185–1205, 2016, https://doi.org/10.1016/j.rser.2016.01.025.

[36] S. Mehta and P. Basak, "Development of PLC-based hardware test-bench prototype for solar-wind-battery-based microgrid system's control algorithm validation," *Electric Power Components and Systems*, pp. 1–20, 2024, https://doi.org/10.1080/15325008.2024.2329326.

[37] O. Dogru *et al*., "Reinforcement Learning in Process Industries: Review and Perspective," in *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 283-300, 2024, https://doi.org/10.1109/JAS.2024.124227.

[38] D. McCarthy *et al*., "A model-based approach to automated validation and generation of PLC code for manufacturing equipment in regulated environments," *Applied Sciences*, vol. 12, no. 15, p. 7506, 2022, https://doi.org/10.3390/app12157506.

[39] A. Mystkowski and A. Kierdelewicz, "Fractional-order water level control based on PLC: hardware-in-the-loop simulation and experimental validation," *Energies*, vol. 11, no. 11, p. 2928, 2018, https://doi.org/10.3390/en11112928.

[40] S. Janarthanan, K. N. Thirukkuralkani and S. Vijayachitra, "Performance analysis of non-integer order PID controller for liquid level control of conical tank system," *International Conference on Information Communication and Embedded Systems (ICICES2014)*, pp. 1-5, 2014, https://doi.org/10.1109/ICICES.2014.7034191.

[41] P. Lanusse and J. Sabatier, "PLC implementation of a CRONE controller," *Fractional Calculus and Applied Analysis*, vol. 14, pp. 505–522, 2011, https://doi.org/10.2478/s13540-011-0031-7.

[42] M. Beschi, F. Padula and A. Visioli, "The generalised isodamping approach for robust fractional PID controllers design," *International Journal of Control*, vol. 90, no. 6, pp. 1157–1164, 2015, https://doi.org/10.1080/00207179.2015.1099076.

[43] A. Kumar and V. Kumar, "Hybridized ABC-GA optimized fractional order fuzzy pre-compensated FOPID control design for 2-DOF robot manipulator," *AEU-international journal of electronics and communications*, vol. 79, pp. 219–233, 2017, https://doi.org/10.1016/j.aeue.2017.06.008.

[44] B. Puchalski, T. A. Rutkowski and K. Duzinkiewicz, "Implementation of the FOPID Algorithm in the PLC Controller - PWR Thermal Power Control Case Study," *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, pp. 229-234, 2018, https://doi.org/10.1109/MMAR.2018.8485807.

[45] A. S. Reddy and G. N. Reddy, "Novel applications of oustaloup recursive approximation and modified oustaloup recursive approximation methods in linear fractional order control design," *International Journal of Dynamics and Control*, vol. 12, no. 7, pp. 2236–2246, 2024, https://doi.org/10.1007/s40435-023-01351-x.

[46] A. Oustaloup, F. Levron, B. Mathieu and F. M. Nanot, "Frequency-band complex noninteger differentiator: characterization and synthesis," in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 1, pp. 25-39, 2000, https://doi.org/10.1109/81.817385.

[47] F. Padula and A. Visioli, "A General Design Methodology for Fractional Cascade Control Systems," in *IEEE Access*, vol. 12, pp. 11451-11457, 2024, https://doi.org/10.1109/ACCESS.2024.3355592.

[48] J. C. C. Basilio, T. R. Oliveira, J. G. T. Ribeiro and A. Cunha, "Evaluation of Fractional-Order Sliding Mode Control Applied to an Energy Harvesting System," *2022 16th International Workshop on Variable Structure Systems (VSS)*, pp. 243-248, 2022, https://doi.org/10.1109/VSS57184.2022.9902105.

[49] M. Kóver-Dorčo and I. Petráš, "Creation of fractional-order $PI^{\lambda}D^{\mu}$ controller function block in Automation Studio environment," *Proceedings of the 2015 16th International Carpathian Control Conference (ICCC)*, pp. 251-254, 2015, https://doi.org/10.1109/CarpathianCC.2015.7145083.

[50] K. Oprzedkiewicz, E. Gawin and T. Gawin, "Real-Time PLC Implementations of Fractional Order Operator," *Automation 2018*, vol. 743, pp. 36–51, 2018, https://doi.org/10.1007/978-3-319-77179-3_4.

[51] K. Oprzedkiewicz *et al.*, "The Frequency and Real-Time Properties of the Microcontroller Implementation of Fractional-Order PID Controller," *electronics*, vol. 10, no. 5, p. 524, 2021, https://doi.org/10.3390/electronics10050524.

[52] A. A. Jamil *et al.*, "Fractional-order PID controllers for temperature control: A review," *Energies*, vol. 15, no. 10, p. 3800, 2022, https://doi.org/10.3390/en15103800.

[53] M. Tavakoli-Kakhki, M. Haeri and M. S. Tavazoei, "Study on Control Input Energy Efficiency of Fractional Order Control Systems," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 3, pp. 475-482, 2013, https://doi.org/10.1109/JETCAS.2013.2273855.

[54] E. Zafiriou and M. Morari, "Design of robust digital controllers and sampling-time selection for SISO systems," *International Journal of Control*, vol. 44, no. 3, pp. 711–735, 1986, https://doi.org/10.1080/00207178608933629.

[55] Jie Chen, S. Hara and Gang Chen, "Best tracking and regulation performance under control energy constraint," in *IEEE Transactions on Automatic Control*, vol. 48, no. 8, pp. 1320-1336, 2003, https://doi.org/10.1109/TAC.2003.815012.

[56] S. Ghasemi, A. Tabesh and J. Askari-Marnani, "Application of Fractional Calculus Theory to Robust Controller Design for Wind Turbine Generators," in *IEEE Transactions on Energy Conversion*, vol. 29, no. 3, pp. 780-787, 2014, https://doi.org/10.1109/TEC.2014.2321792.

[57] S. Katoch, S. S. Chauhan and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia tools and applications*, vol. 80, pp. 8091-8126, 2021, https://doi.org/10.1007/s11042-020-10139-6.

[58] L. Y. Chang and H. C. Chen, "Tuning of fractional PID controllers using adaptive genetic algorithm for active magnetic bearing system," *WSEAS Transactions on systems*, vol. 8, no. 1, pp. 158-167, 2009, https://www.wseas.us/e-library/transactions/systems/2009/31-894.pdf.

[59] H. Bouyedda and S. Ladaci, "Optimal tuning of fractional order $PI^{\lambda}D^{\mu}$ controllers using genetic algorithms," *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 207-212, 2016, https://doi.org/10.1109/ICMIC.2016.7804299.

[60] A. Lambora, K. Gupta and K. Chopra, "Genetic Algorithm- A Literature Review," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 380-384, 2019, https://doi.org/10.1109/COMITCon.2019.8862255.

[61] Zwe-Lee Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," in *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 384-391, 2004, https://doi.org/10.1109/TEC.2003.821821.

[62] M. Zamojski, "Implementation of fractional order PID controller based on recursive oustaloup'e filter," *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pp. 414-417, 2018, https://doi.org/10.1109/IIPHDW.2018.8388402.

[63] J. Baranowski, W. Bauer, M. Zagórowska, T. Dziwinski and P. Piatek, "Time-domain Oustaloup approximation," *2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 116-120, 2015, https://doi.org/10.1109/MMAR.2015.7283857.

[64] A. Oustaloup, F. Levron, B. Mathieu and F. M. Nanot, "Frequency-band complex noninteger differentiator: characterization and synthesis," in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 1, pp. 25-39, 2000, https://doi.org/10.1109/81.817385.

[65] A. Tepljakov, "FOMCON toolbox for modeling, design and implementation of fractional-order control systems," *Applications in control*, vol. 6, pp. 211-236, 2019, https://doi.org/10.1515/9783110571745.

[66] C. Wang, G. Yin, C. Liu and W. Fu, "Design and simulation of inverted pendulum system based on the fractional PID controller," *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1760-1764, 2016, https://doi.org/10.1109/ICIEA.2016.7603871.

[67] K. Oprzedkiewicz and W. Mitkowski, "Parameter Identification for Non Integer Order, Discrete, State Space Model of Heat Transfer Process Using CFE Approximation," *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, pp. 436-440, 2018, https://doi.org/10.1109/MMAR.2018.8486115.

[68] K. Oprzedkiewicz, E. Gawin and W. Mitkowski, "A plc implementation of PSE approximant for fractional order operator," *Non-Integer Order Calculus and its Applications*, vol. 496, pp. 102–112, 2019, https://doi.org/10.1007/978-3-319-78458-8_10.

[69] F. Padula and A. Visioli, "A General Design Methodology for Fractional Cascade Control Systems," in *IEEE Access*, vol. 12, pp. 11451-11457, 2024, https://doi.org/10.1109/ACCESS.2024.3355592.

[70] Chunna Zhao, Dingyu Xue and YangQuan Chen, "A fractional order PID tuning algorithm for a class of fractional order plants," *IEEE International Conference Mechatronics and Automation*, vol. 1, pp. 216-221, 2005, https://doi.org/10.1109/ICMA.2005.1626550.

[71] N. E. Badau *et al.*, "Bridging the gap between control theory and practice: From simple controller design to a practical microcontroller implementation," *IFAC-PapersOnLine*, vol. 58, no. 26, pp. 124–129, 2024, https://doi.org/10.1016/j.ifacol.2024.10.282.

[72] A. Chevalier *et al.*, "Fractional-order PID design: Towards transition from state-of-art to state-of-use," *ISA transactions*, vol. 84, pp. 178–186, 2019, https://doi.org/10.1016/j.isatra.2018.09.017

[73] S. A. Mehta, D. M. Adhyaru and M. Vadsola, "Comparative study for various fractional order system realization methods," *2013 Nirma University International Conference on Engineering (NUiCONE)*, pp. 1-4, 2013, https://doi.org/10.1109/NUiCONE.2013.6780179.

[74] S. Das, S. Das and A. Gupta, "Fractional order modeling of a PHWR under step-back condition and control of its global power with a robust $PI^{\lambda}D^{\mu}$ controller," in *IEEE Transactions on Nuclear Science*, vol. 58, no. 5, pp. 2431-2441, 2011, https://doi.org/10.1109/TNS.2011.2164422.

[75] P. Shah *et al.*, "Fractional Order Control: A Bibliometric Analysis (2000–2022)," *Results in Control and Optimization*, vol. 14, p. 100366, 2024, https://doi.org/10.1016/j.rico.2023.100366.

[76] N. S. Solke *et al.*, "Machine Learning-Based Predictive Modeling and Control of Lean Manufacturing in Automotive Parts Manufacturing Industry," *Global Journal of Flexible Systems Management*, vol. 23, pp. 89–112, 2022, https://doi.org/10.1007/s40171-021-00291-9.

[77] P. Shah, R. Sekhar and D. Sharma, "Experimental Analysis of Fractional PID Controller Parameters on Time Domain Specifications," *Progress in Fractional Differentiation and Applications*, vol. 3, no. 2, pp. 121-134, 2017, https://doi.org/10.18576/pfda/030205.

[78] S. Jaiswal *et al.*, "Design of Fractional Order PID Controller Using Genetic Algorithm Optimization Technique for Nonlinear System," *Chemical Product and Process Modeling*, vol. 15, no. 2, p. 20190072, 2020, https://doi.org/10.1515/cppm-2019-0072.

[79] X. Liu, "Optimization design on fractional order PID controller based on adaptive particle swarm optimization algorithm," *Nonlinear Dynamics*, vol. 84, pp. 379–386, 2016, https://doi.org/10.1007/s11071-015-2553-8.

[80] R. Mennilli, L. Mazza and A. Mura, "Integrating Machine Learning for Predictive Maintenance on Resource-Constrained PLCs: A Feasibility Study," *Sensors*, vol. 25, no. 2, p. 537, 2025, https://doi.org/10.3390/s25020537.

[81] A. Mystkowski and A. Kierdelewicz, "Fractional-order water level control based on PLC: hardware-in-the-loop simulation and experimental validation," *Energies*, vol. 11, vol. 11, p. 2928, 2018, https://doi.org/10.3390/en11112928.

[82] M. Dhasagounder *et al.*, "An Application of Artificial Bee Colony and Cohort Intelligence in the Automatic Generation Control of Thermal Power Systems," *Intelligent Methods in Electrical Power Systems*, pp. 23–41, 2024, https://doi.org/10.1007/978-981-97-5718-3_2.

[83] P. Shah R. Sekhar and S. Agashe, "Application of fractional PID controller to single and multi-variable non-minimum phase systems," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 2801-2811, 2019, https://www.researchgate.net/profile/Ravi-Sekhar/publication/338343086_Application_of_Fractional_PID_Controller_to_Single_and_Multi-Variable_Non-Minimum_Phase_Systems/links/5e0e1666299bf10bc38c0c38/Application-of-Fractional-PID-Controller-to-Single-and-Multi-Variable-Non-Minimum-Phase-Systems.pdf.

[84] S. Francis, P. Shah, R. Sekhar and A. P. Singh, "Fractional Order PID Controller Design and Tuning via Oustaloup Approximation for TCLab System," *2025 International Conference on Intelligent Control, Computing and Communications (IC3)*, pp. 887-891, 2025, https://doi.org/10.1109/IC363308.2025.10956243.