fractal and fractional

MDPI

Article

Hybrid PSO-Tuned Fractional-Order Control with Rule-Based Adaptive Supervision for Embedded Thermoelectric Temperature Regulation

Miguel F. Ferrer Pareja [1], Carlos Sánchez Morales [2], Federico León Zerpa [2] and Alejandro Ramos Martín [1,*]

[1] University Institute of Intelligent Systems and Numerical Applications in Engineering, University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain; miguel.ferrer104@alu.ulpgc.es
[2] Department of Process Engineering, University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain; carlos.sanchez@ulpgc.es (C.S.M.); federico.leon@ulpgc.es (F.L.Z.)
* Correspondence: alejandro.ramos@ulpgc.es; Tel.: +34-928-451-933

Abstract

Thermal regulation using Peltier cells presents challenges due to high inertia, memory effects, and energy constraints in embedded systems. This paper introduces the FOPID with Adaptive Supervisor (FOPID-AS) scheme, combining a PSO-optimized fractional-order controller (FOPID) with a deterministic rule-based gain-scheduling supervisor. Experimental validation compares four strategies: PID, Fuzzy-PID, static FOPID, and the proposed FOPID-AS. During the transient phase ($t < 105$ s), FOPID-AS reaches the ± 0.5 °C tolerance band in 31.20 s, with an ITAE of 6612.97 and transient energy consumption of 0.18 Wh, outperforming PID, Fuzzy-PID, and FOPID in speed and tracking quality. In steady state ($t \geq 105$ s), FOPID-AS exhibits steady-state error $e_{ss} = 0.08$ °C, $\sigma_{ss} = 0.10$ °C, and peak-to-peak ripple of 0.67 °C, with steady-state energy consumption of 0.30 Wh, showing lower dispersion than PID and comparable values to the other fractional controllers, while maintaining low computational load suitable for real-time applications.

Keywords: fractional control; FOPID; rule-based gain scheduling; Peltier; thermal regulation; embedded systems

1. Introduction

High-precision thermal regulation in condensation-sensitive systems poses a challenge due to the limitations of traditional PID controllers. Specifically, the derivative term in PID amplifies high-frequency measurement noise, generating rapid and large control variations that may damage actuators or induce saturation [1]. Moreover, increasing the integral gain to reduce steady-state error can lead to sustained oscillations and reduced stability margins, especially in thermal systems where sensor–actuator delays require careful anticipation of the error to avoid excessive temperature fluctuations.

Classical PID controllers also have structural limitations: their integer-order nature restricts achievable magnitude and phase relationships, making it difficult to meet precise frequency-domain performance objectives [2]. In practice, the derivative term is often filtered or omitted, while the integral term introduces phase lag that further reduces stability [3]. These constraints demonstrate that PID controllers may be insufficient for high-accuracy thermal regulation in Peltier-based systems, motivating the development of

Academic Editor: David Kubanek

Received: 28 February 2026
Revised: 27 March 2026
Accepted: 30 March 2026
Published: 3 April 2026

alternative methods capable of estimating and rejecting the total disturbance, including unmodeled dynamics and external perturbations.

The approach proposed in this work combines a fractional-order PID (FOPID) with a rule-based gain-scheduling supervisor that updates controller parameters in real time according to bounded error regions, keeping the Peltier cell temperature close to the set-point T_{ref} under load variations. Fractional controllers' higher sensitivity to parametric variations [4] motivates a hybrid architecture separating offline optimization from online adaptation using deterministic rules [5]. Experimental validation uses a structured comparative protocol with multiple performance metrics, including ITAE, energy consumption, thermal ripple, and kernel probability density estimation (KDE), allowing comprehensive quantitative characterization against PID, FOPID, and FOPID-AS strategies. The deterministic supervision structure avoids online learning, ensuring limited and predictable computational complexity suitable for embedded platforms [6].

2. State of the Art

2.1. HeatTransfer in Peltier Cells

The thermal dynamics in Peltier cells are characterized by high inertia and memory effects associated with the non-local diffusion of heat in their semiconductor materials. As illustrated in Figure 1, the thermal flow generates an experimental response that does not follow a classical exponential relaxation, but rather exhibits a long-range dependence on previous thermal states. This direct link between thermal memory and the observed experimental behavior is modeled with high fidelity using fractional formulations.

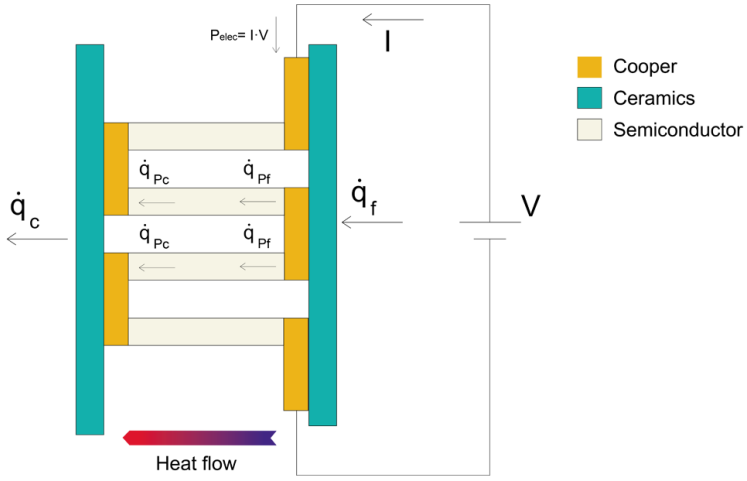

Figure 1. Diagram of the thermoelectric device (Peltier cell). The structure of ceramics, copper, and semiconductors is illustrated, as well as the heat flow induced by electrical power.

Various studies have shown that these thermal memory effects can be described compactly using these formulations, which allow the long-range temporal dependence observed experimentally in thermoelectric systems to be captured without resorting to distributed models [7,8] with explicit spatial dependence. These models allow the anomalous diffusion phenomenology observed in laboratory tests to be captured in a compact form, avoiding the computational complexity of full distributed models.

2.2. First-Order Fractional Model

For the design of the control system, the thermal dynamics are approximated by a first-order fractional input–output model. The model is based on the widely studied fractional relaxation equation [9], expressed as:

$$\tau D^{\gamma} y(t) + y(t) = f(t), \tag{1}$$

where D^{γ} denotes a fractional derivative of order $0 < \gamma \leq 1$ and τ represents a generalized time constant.

Applying the Laplace transform and assuming zero initial conditions, the input–output relationship in the frequency domain is obtained:

$$G(s) = \frac{Y(s)}{F(s)} = \frac{1}{1 + \tau s^{\gamma}}. \tag{2}$$

In practical applications of control and experimental identification, a static gain K is introduced to adjust the level of the system response, giving rise to the first-order fractional model used in this work:

$$G(s) = \frac{K}{1 + \tau s^{\gamma}}. \tag{3}$$

In this model, K represents the static gain, γ the fractional order of the system, and τ a quasi time constant characteristic of the fractional dynamics, rather than a classical time constant in seconds. The input $f(t)$ is assumed to be bounded and continuous, and the system is considered to start from zero initial condition. This formulation provides a compact and adequate representation of the experimentally observed thermal behavior, allowing thermal memory and non-local diffusion effects to be captured without resorting to a complete distributed model with spatial dependence. For the controller design, the thermal dynamics are approximated by a first-order fractional input–output model, which has been shown to adequately capture memory effects and non-local diffusion in real thermal systems with a reduced number of parameters [10]. The model parameters $\{K, \tau, \gamma\}$ are identified experimentally using a least-squares fitting procedure applied to the step response of the system. To ensure the fidelity of the identification in the face of unmodeled dynamics, a time window selection is made that spans from the start of the stimulus to stabilization in steady state, incorporating noise rejection techniques through prior digital filtering. In thermal regulation applications using Peltier cells, the fractional order γ typically ranges from $0.85 \leq \gamma \leq 0.95$, accurately capturing the sub-diffusive nature and thermal memory effects observed [9]. This compact representation is essential for the design of controllers in embedded platforms with limited resources.

The fractional-order model was fitted to both the experimental data and a three-node thermal model. The identified parameters ($K = 3.873$, $\tau = 20$ s, $\gamma = 1.177$) characterize the dynamic response of the system under the applied cooling conditions. As shown in Table 1 and Figure 2, the model reproduces the temperature evolution of the cold node with high fidelity.

Table 1. Fractional-order model fitting results by transient phase.

Phase	K	τ (s)	γ	R^2 (3-Node)	RMSE (°C)
transition	1.929	15.802	5.936	0.99	0.87
fine	0.342	1142.800	2.018	0.98	0.92

The coefficient of determination reached $R^2 = 0.9991$ with respect to the three-node model and $R^2 = 0.9938$ with respect to the experimental measurements, while the root mean square error (RMSE) remained below 1 °C in both cases. The fractional order $\gamma > 1$ indicates super-diffusive behavior, consistent with the combined conduction and Peltier-induced advection effects, and reflects the accelerated thermal propagation observed in the Peltier module. These results confirm that the fractional-order formulation provides a

compact representation of the system dynamics while preserving agreement with both the physics-based model and the measured data.

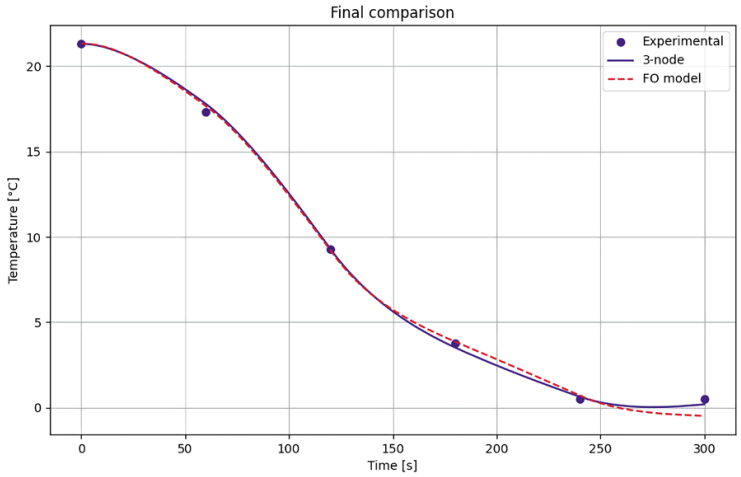

Figure 2. Comparison of the temperature from experimental measurements, the three-node thermal model, and the fitted fractional-order model.

To describe the physical significance of the fractional-order parameter γ, the transient response was divided into two phases based on the absolute control error: transition and fine. The aggressive phase was excluded due to rapidly varying errors that hinder reliable model fitting. Fractional-order (FO) models were fitted within each phase, providing phase-specific identification results. As shown in Table 2, the approximately constant γ values indicate that it reflects intrinsic thermal properties of the Peltier system. Specifically, γ quantifies the non-exponential, memory-like behavior of heat transport, arising from both material conduction and Peltier-induced advective effects.

Table 2. Fractional-order model parameters per phase.

Phase	K	τ (s)	γ
transition	1.93	15.80	5.94
fine	0.34	1142.80	2.02

2.3. FOPID Controller

FOPID controllers extend classic PID controllers by including integrals and derivatives of fractional order λ and μ. The resulting fractional control action is defined as:

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) + K_{ff} f(e(t)) \quad (4)$$

where $e(t) = T_{ref} - T_{sup}$ represents the temperature error.

The first three terms correspond to the standard FOPID described by Podlubny [9], whose flexibility has allowed the development of optimal tuning rules aimed at minimizing the integrated absolute error under robustness constraints [11]. The additional term $K_{ff} f(e(t))$ is introduced as a rule-based gain-scheduled extension, which adjusts the control action in real time to improve performance in the face of disturbances and nonlinearities. This structure allows for greater flexibility and guarantees robustness specifications in the face of gain variations and high-frequency noise [12].

For its digital implementation in embedded platforms, this work adopts the Grünwald–Letnikov approximation with finite memory M, due to its computational simplicity and suitability for discrete-time systems [13]:

$$D^{\alpha}e[n] = T_s^{-\alpha} \sum_{j=0}^{\min(n,M)} (-1)^j \binom{\alpha}{j} e[n-j] \tag{5}$$

with coefficients calculated recursively as:

$$w_0^{(\alpha)} = 1, \quad w_k^{(\alpha)} = w_{k-1}^{(\alpha)} \frac{\alpha - k + 1}{k}, \quad k = 1, \ldots, M. \tag{6}$$

The short memory principle allows the sum to be reduced to the last M terms, adapting the computational complexity to limited hardware. The weights are precalculated to minimize the load during the control loop. In the embedded implementation, the upper limit of the summation is truncated to the last M samples according to the finite-memory Grünwald–Letnikov principle. Here, M is the selected memory length, set to 30 in the experiments.

The accuracy of this numerical approximation depends critically on refinement, with implementation errors being minimized as the non-local nature of the derivative intensifies at values less than α. For implementation on the embedded platform (Arduino), a memory length $M = 30$ and a sampling period $T_s = 0.1$ s (value adapted to the thermal inertia of the system) are selected. Following the Short Memory (L_m) principle, this truncation mitigates the computational load associated with the non-local nature of fractional operators [13]. A critical aspect of the proposed architecture is that the controller parameters are non-stationary: the adaptive supervisor modifies the orders λ and μ as well as the feedforward and proportional gains in real time based on the thermal error [14]. Consequently, the Grünwald–Letnikov weights w_k are updated online using the initGL() function whenever the supervisor detects parameter changes, ensuring that the fractional approximation accurately reflects the requested dynamics at each moment of execution [15]. This dynamic adaptation allows the controller to respond effectively to disturbances and varying operating conditions while maintaining stability and precision.

Table 3 summarizes the effect of the short-term memory length M on control performance. An intermediate memory length M balances responsiveness and noise attenuation, achieving the best trade-off between transient performance and stability. Metrics such as ITAE, IAE, RMSE, and steady-state error (σ_{ss}) are optimized at this setting, while maintaining low computational cost in terms of CPU usage. The sampling period T_s was selected to capture the fast thermal dynamics of the system while keeping the computational load within practical limits. This discretization allows effective control of the transient response, whereas longer sampling periods reduce accuracy and shorter periods increase computational demand.

Table 3. Effect of short-term memory length M on control performance.

M	ITAE	IAE	RMSE	σ_{ss} (°C)	CPU (%)	t_{rec} (s)
20	12,973.487	31.416	0.190	0.151	45.325	35.53
30	**11,261.408**	**24.632**	**0.157**	**0.136**	**32.635**	**34.91**
40	12,967.707	31.281	0.190	0.150	33.356	36.41

Additionally, $M = 30$ yields the lowest computational cost in terms of CPU usage, while maintaining comparable loop execution time and memory consumption. In contrast, both shorter ($M = 20$) and longer ($M = 40$) memory horizons result in degraded performance, suggesting that an intermediate memory length effectively balances responsiveness and noise attenuation. These results highlight the existence of an optimal memory horizon for achieving robust and efficient control.

2.4. Rule-Based Gain-Scheduling Supervisor

The rule-based gain-scheduling supervisor divides the operating space into three well-defined regions according to the absolute value of the error $|e(t)|$. This logic, implemented in an external Python 3.10.11 service communicated via TCP with the visualization system, allows for dynamic reconfiguration of the FOPID parameters and the feedforward term [15].

In particular, the supervisor classifies the error using the thresholds e_{low} and e_{high}. For large errors, the proportional gain K_p is increased and the fractional orders are reduced to improve stability and response speed. For moderate errors, the gains are adjusted gradually using linear interpolation. Finally, near the setpoint, a feedforward term K_{ff} is activated to minimize error variance and thermal ripple. The decision rules are shown in the region map in Table 4.

Table 4. Adaptive supervision rules according to error regions.

Region	Condition	$\mathbf{K_p}$	λ	μ	$\mathbf{K_{ff}}$
Recovery	$\|e\| > 3.0$	$1.3\,K_{p,nom}\,A$	0.55	1.10	0
Transition	$0.3 \leq \|e\| \leq 3.0$	Interpolated	Interpolated	Interpolated	Interpolated
Accuracy	$\|e\| < 0.3$	$0.8\,K_{p,nom}$	0.85	1.85	$0.08\,T_{ref}$

To manage the thermal asymmetry of the Peltier cell, an aggressiveness factor ($A = 1.2$ if $e > 0$, otherwise $A = 1.0$) is introduced, increasing the proportional gain during heating slopes. In the Precision Region, defined by a hysteresis band of ±0.3 °C, the controller reduces K_p and increases the orders (λ, μ) to maximize smoothness and noise filtering, minimizing e_{ss} (steady-state error) [14]. In the intermediate zone, linear interpolation based on a weighting factor is applied:

$$f = \frac{|e| - 0.3}{3.0 - 0.3} \tag{7}$$

This scheme ensures a smooth transition between operating modes, avoiding abrupt switching that could induce instability in the embedded platform [13]. The lower threshold (0.3 °C) was chosen based on the sensor precision and typical noise fluctuations, defining the range in which fine adjustments are sufficient to maintain stability and minimize thermal ripple. The upper threshold (3 °C) corresponds to the error level beyond which aggressive correction is required to quickly return to the setpoint while preserving system stability. The intermediate region interpolates controller parameters to ensure smooth transitions. These thresholds were validated through simulation and experimental testing, balancing recovery speed, stability, and smoothness of the control signal.

2.5. Stability Considerations

The proposed supervisory mechanism does not rely on recursive parameter adaptation or identification schemes. Instead, controller parameters are updated according to predefined bounded regions determined by the instantaneous tracking error. All controller gains and fractional orders remain explicitly constrained within fixed intervals derived from the supervisory rules. Since the nominal FOPID configuration was experimentally validated to ensure closed-loop stability, and since the scheduling mechanism does not introduce additional adaptation dynamics, the overall system preserves practical Bounded-Input, Bounded-Output stability under bounded tracking error conditions.

A sensitivity analysis of the aggressiveness factor A was conducted to evaluate its impact on the transient performance of the FOPID-AS controller. The parameter was tested in the range 1.0–1.3 under demanding conditions (T_{ref} = 8 °C). While the transient response is limited by actuator saturation, steady-state analysis shows that $A = 1.2$ provides the

best trade-off in terms of accuracy, thermal stability, and control effort, as shown in Table 5. Therefore, this value was selected as the nominal setting. Metrics were computed after the system reached steady state to avoid actuator saturation effects.

Table 5. Steady-state performance analysis at 8 °C for different values of A.

A	ITAE	IAE	$\mathbf{RMSE}_{ss}$	σ_{ss} (°C)	$\mathbf{PWM}_{mean,ss}$
1.0	31,833.33	0.65	0.65	0.05	121.12
1.1	35,962.00	0.69	0.69	0.03	122.88
1.2	32,202.38	0.61	0.61	0.03	119.34
1.3	32,469.29	0.64	0.64	0.05	120.73

3. Methodology

Control Architecture

The complete system architecture includes three layers, as shown in Figure 3.

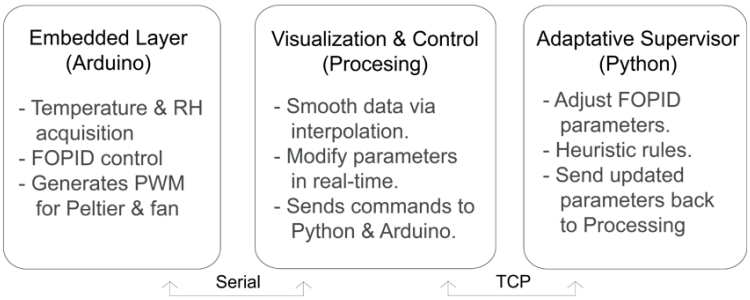

Figure 3. FOPID hybrid control architecture with rule-based adaptive supervisor. Serial and TCP flows between Arduino (IDE v2.3.8), Processing (v4.5.2), and Python (v3.10.11) are shown.

The control system is implemented using a distributed hybrid architecture with three levels: an embedded level in Arduino for signal acquisition, FOPID execution, and actuation on the Peltier cell; a supervisor in Python that dynamically adjusts parameters in real time following deterministic rules; and a visualization and communication level for monitoring and interaction. The surface temperature is measured with an NTC thermistor [16], filtered using a discrete exponential filter, and the ambient temperature and humidity are obtained with an SCD41 sensor to estimate the dew point and ensure safe operation. The control signal is applied as PWM limited by a smooth ramp, while the supervisor divides the operating space into three regions according to the thermal error, adjusting proportional, integral, derivative, and applying feed-forward in a regime close to the reference. This methodology combines fractional control with light deterministic adaptation, ensuring accuracy, stability, and energy efficiency in systems with high inertia and delays without increasing the computational load.

4. Optimization Using PSO

FOPID optimization is performed offline using PSO, a low-computational-cost strategy that performs well compared to conventional methods. This technique, in line with computer-aided design tools [17], has been used to tune fractional controllers under criteria such as ITAE and ISE [18,19], instead of step-based heuristics.

The tuning of the FOPID parameters, $\boldsymbol{\theta} = [K_p, K_i, K_d, \lambda, \mu]$, is performed offline using a multi-objective PSO, considering dynamic performance and energy efficiency, obtaining a nominal set suitable for embedded systems.

The objective of the optimization is to minimize a composite criterion that simultaneously integrates the dynamic performance of the controller and its energy efficiency. The objective function used is defined as:

$$J = w_1 \,\mathrm{ITAE} + w_2 \,\mathrm{ISE} + w_3 \, E_{\mathrm{cons}} + w_4 \, \sigma_T \tag{8}$$

where the coefficients w_i are predefined constants that combine the different performance indices into a single scalar objective function. These coefficients are selected to ensure a balanced contribution of transient response, error magnitude, energy consumption, and thermal variability, without introducing additional normalization constraints. The term $\mathrm{ITAE} = \int_0^{t_f} t\,|e(t)|\,dt$ penalizes persistent errors; $\mathrm{ISE} = \int_0^{t_f} e^2(t)\,dt$ penalizes instantaneous errors of large magnitude; E_{cons} represents the total energy consumed by the Peltier cell during the test; and σ_T quantifies the thermal variability with respect to the reference using the standard deviation. This formulation is based on similar strategies reported in [2,11,15] for the optimization of FOPID fractional controllers.

The tuning of the FOPID parameters is performed using the Particle Swarm Optimization (PSO) algorithm, configured with a global swarm topology (gbest) and uniform random initialization. For the evaluation of fractional dynamics within optimization, a memory window $M = 20$ is used. The search space limits and swarm configuration are detailed in Table 6.

Table 6. Technical configuration and PSO search ranges.

Configuration	Value/Range
Swarm size (*nParticles*)	30
Maximum iterations (*nIter*)	50
Simulation step (dt)	0.5 s
GL memory in optimization (M)	20
Proportional gain range (K_p)	$[10, 150]$
Gain range K_i, K_d	$[0, 5]$
Order range λ, μ	$[0.5, 1.5]$

This heuristic search process allows us to identify the optimal parameter vector $\theta = [K_p, K_i, K_d, \lambda, \mu]$ that minimizes the cumulative error, ensuring a robust basis for subsequent adaptive supervision in the real system. Table 7 shows the optimal values obtained.

Table 7. Optimal FOPID parameters after PSO optimization.

$\mathbf{K_p}$	$\mathbf{K_i}$	$\mathbf{K_d}$	$\boldsymbol{\lambda}$	$\boldsymbol{\mu}$
58.93	3.91	2.66	0.67	1.47

These parameters allow the FOPID to control the Peltier cell with a smoother response, lower cumulative error, and reduced energy consumption compared to a classic PID or hysteresis control.

4.1. Sensitivity Analysis of the Optimization Process

A sensitivity analysis was conducted to evaluate the influence of key PSO configuration parameters, including swarm size and maximum number of iterations, as well as the weighting factors of the objective function. The results are summarized in Table 8.

Variations in the PSO configuration lead to moderate changes in the optimized FOPID parameters, particularly in the integral and derivative gains, as well as in the fractional orders. Similarly, ±10% perturbations in the objective function weights result in different parameter combinations.

However, these variations do not alter the overall structure of the solution, but rather indicate the existence of multiple parameter sets capable of achieving comparable control performance. This suggests the presence of a relatively flat optimum region in the search space, supporting the robustness of the tuning process with respect to moderate perturbations in both the optimization algorithm and the objective function.

Table 8. Sensitivity analysis of PSO configuration and objective function weights.

Configuration	Optimized Parameters [Kp, Ki, Kd, λ, μ]
20 particles, 40 iterations	[58.94, 4.55, 3.44, 0.70, 1.02]
30 particles, 50 iterations	[**58.93**, **3.91**, **2.66**, **0.67**, **1.47**]
40 particles, 100 iterations	[58.93, 4.64, 4.96, 0.72, 0.50]
$w_1 = 1.0, w_2 = 0.1, w_3 = 0.01$	[54.85, 5.00, 5.00, 0.50, 0.67]
$w_1 = 1.1, w_2 = 0.1, w_3 = 0.01$	[54.86, 1.26, 5.00, 0.53, 1.08]
$w_1 = 0.9, w_2 = 0.1, w_3 = 0.01$	[54.86, 1.20, 4.20, 0.85, 1.36]
$w_1 = 1.0, w_2 = 0.11, w_3 = 0.01$	[54.87, 4.47, 4.90, 0.50, 0.76]
$w_1 = 1.0, w_2 = 0.1, w_3 = 0.011$	[54.86, 4.15, 4.12, 1.49, 1.28]

4.2. Reproducibility and Repeatability of Experimental Trials

The controller's performance was evaluated by separating transient and steady state, with a threshold of 105 s defined by the PID settling time at 5% tolerance. During the transient, undershoot, band entry time $t_{\pm 0.5}$, and ITAE (ITAE $= \int_0^{105} t\,|e(t)|\,dt$) were considered, as well as energy consumption and PWM peak. In steady state ($t \geq 105$ s), the mean error, steady-state standard deviation σ_{ss}, peak-to-peak ripple, and energy consumption were evaluated, quantifying accuracy, stability, and thermal efficiency.

The repeatability of each control strategy was assessed using the mean surface temperature $\overline{T}_{sup}$ measured in each trial for $t \geq 105$ s and the standard deviation between trials σ_{rep}. This analysis provides a measure of the experimental reproducibility of the system once steady state is reached. According to the results, all controllers maintain surface temperatures close to the reference value of 12 °C, with low variability between repetitions. The lowest coefficients of variation are observed for PID (0.10%) and FOPID-AS (0.07%), while Fuzzy-PID and FOPID show higher variability, reflecting the differences in steady-state reproducibility across trials, as shown in Table 9.

Table 9. Reproducibility of steady-state surface temperature between repetitions.

Control	Average (°C)	σ_{rep} (°C)	CV (%)
PID	12.3356	0.0127	0.10
Fuzzy-PID	12.3759	0.0208	0.17
FOPID	12.0123	0.0357	0.30
FOPID-AS	11.9709	0.0084	0.07

An analysis of variance (ANOVA) was conducted to assess whether the differences among controllers were statistically significant, followed by a Tukey HSD post hoc test for pairwise comparisons. Results are shown in Table 10.

The ANOVA revealed significant differences among controllers ($F = 45.18$, $p < 0.001$). The Tukey HSD test indicates that FOPID-AS is significantly more accurate than PID, Fuzzy-PID, and FOPID, while FOPID and FOPID-AS do not differ significantly. This confirms that FOPID-AS provides the best reproducibility and thermal stability in steady state.

Table 10. ANOVA and Tukey HSD for steady-state surface temperature T_{sup}.

Comparison	Mean Difference (°C)	Adjusted p-Value	Significant?
FOPID vs. FOPID-AS	−0.1005	0.3260	No
FOPID vs. Fuzzy-PID	0.4456	0.0002	Yes
FOPID vs. PID	0.3298	0.0014	Yes
FOPID-AS vs. Fuzzy-PID	0.5460	0.0000	Yes
FOPID-AS vs. PID	0.4302	0.0002	Yes
Fuzzy-PID vs. PID	−0.1158	0.2275	No

5. Results

5.1. Robustness and Iso-Damping Analysis

The designed system exhibits superior robustness through the iso-damping property, which is analytically based on the open-loop frequency response. This characteristic is defined by the existence of a flat phase in the vicinity of the crossover frequency, where the slope of the phase with respect to frequency is practically zero ($d\phi/d\omega \approx 0$) [2].

This topology ensures that the phase margin remains virtually constant in the face of fluctuations in the static gain of the plant, which are inherent in Peltier cells due to thermal and load variations. Consequently, following the robustness criteria set forth by Padula and Visioli [11], the overshoot of the gain-scheduled fractional controller remains robust and predictable within the bounded parameter region. This mitigates the risks of thermal instability in the face of external disturbances that often degrade the performance of conventional full-order controllers [20].

5.2. Statistical Analysis of Accuracy and Error Distribution

To isolate the impact of each architectural component, an ablation study was performed comparing three successive stages: the base controller (PID), the introduction of fractional dynamics (FOPID), and the complete system with adaptive supervision (FOPID-AS) [15]. Steady-state accuracy was evaluated using a Kernel Density Estimation (KDE) model, using a Gaussian kernel and a bandwidth optimized using Silverman's rule to ensure an accurate representation of the error distribution. From a control perspective, this analysis provides a probabilistic characterization of the steady-state error, allowing the assessment of its dispersion and the likelihood of deviations from the reference. As can be seen in Figure 4, FOPID-AS shows the highest leptokurtosis, confirming that the combination of fractional memory and rule-based adaptation based on the scheme of Yin et al. [15] is the optimal configuration for confining the error to a narrow band close to zero [14].

The detailed behavior shows that the proposed FOPID-AS scheme achieves a significantly higher error concentration in the vicinity of $e(t) = 0$. While conventional PID and FOPID have wider and more platykurtic distributions—indicating greater sensitivity to disturbances—the fractional adaptive strategy confines the error to a narrow band, reducing the variance of the system. From a control standpoint, this concentration around zero implies improved steady-state precision and a lower probability of large deviations, which are directly related to enhanced robustness under small perturbations. This higher probability density at the origin confirms the superior robustness and regulatory accuracy of the FOPID-AS controller under real operating conditions. Additional quantitative measures of the error distribution, including standard deviation, kurtosis, and 95% width, are reported in Table 11, complementing the KDE analysis. These results confirm that FOPID-AS confines the error near zero more effectively than the other controllers.

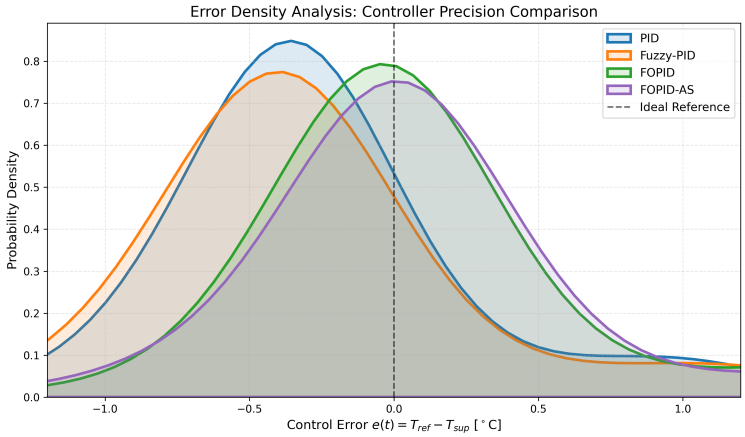

Figure 4. Probability density distribution of the control error. The green curve (FOPID-AS) exhibits greater leptokurtosis and a higher zero-centeredness compared to the conventional PID (blue) and static FOPID (orange).

Table 11. Statistical metrics of the steady-state error distribution for different controllers.

Controller	Std of Error (°C)	Kurtosis	Width 95% (°C)
PID	1.61	8.57	8.10
Fuzzy-PID	1.84	6.99	9.00
FOPID	1.76	7.24	8.86
FOPID-AS	1.76	6.45	8.97

5.3. Analysis of Parameter Adaptation Dynamics

Unlike classical PID and static FOPID controllers, where parameters remain fixed throughout the operation, the proposed FOPID-AS scheme introduces a time-varying parameter adaptation mechanism. This difference is particularly relevant in the transient regime, where fixed-parameter controllers exhibit less flexibility to respond to disturbances and nonlinear thermal effects. The robustness of the proposed controller to thermal nonlinearities is based on the supervisor's reconfiguration capability. As shown in Figure 5, the fractional orders (λ, μ) and the proportional gain (K_p) exhibit a dynamic evolution synchronized with the control error, allowing a smooth transition between the transient regime and the steady state, demonstrating that the supervisor optimizes system performance by increasing the derivative order μ during heating slopes, which introduces active damping. As the temperature stabilizes at the reference value, the parameters converge to values that minimize the error variance, ensuring precise regulation. This adaptation mechanism explains the statistical superiority previously observed in the KDE error distributions.

The analysis of parameter adaptation shows that K_p, λ, μ, and K_{ff} increase when the system requires correction and decrease as the temperature approaches the setpoint, which explains why the evolution plots display smooth transitions and why the error remains confined near zero. The positive correlation of λ, μ, and K_{ff} with the control error indicates that the supervisor adjusts multiple aspects of the controller simultaneously to absorb disturbances and maintain stability. K_p, λ, μ, and K_{ff} exhibit clear, correlated variations with the error, confirming that the supervisor responds immediately to perturbations. In contrast, K_i and K_d remain constant, serving primarily to maintain long-term stability rather than contributing to instantaneous adaptation. This behavior contrasts with the fixed dynamics observed in PID and FOPID controllers, where the lack of parameter adaptation limits the ability to react to changing operating conditions.

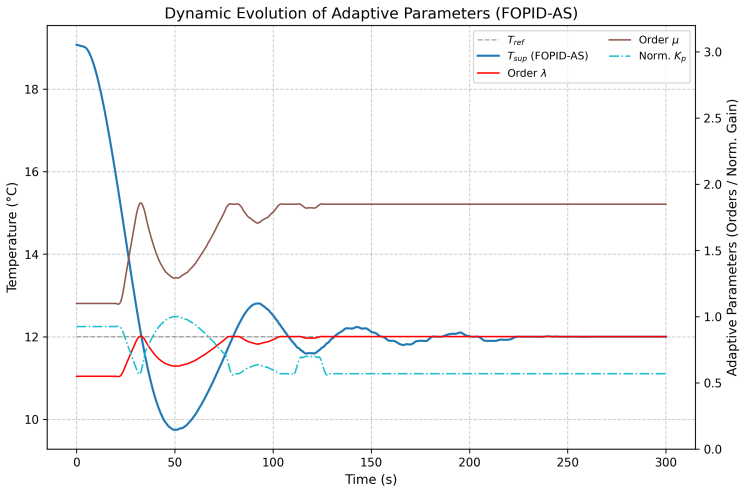

Figure 5. Temporal evolution of the thermal response and adaptive parameters of the FOPID-AS controller. In contrast to the fixed-parameter behavior of PID and static FOPID controllers, the dynamic adjustment of the fractional orders enables real-time compensation of disturbances and smoother convergence.

5.4. Reference Tracking

The hybrid controller keeps T_{sup} close to T_{ref} with minimal oscillation and fast settling. Figure 6 shows an example of reference tracking.

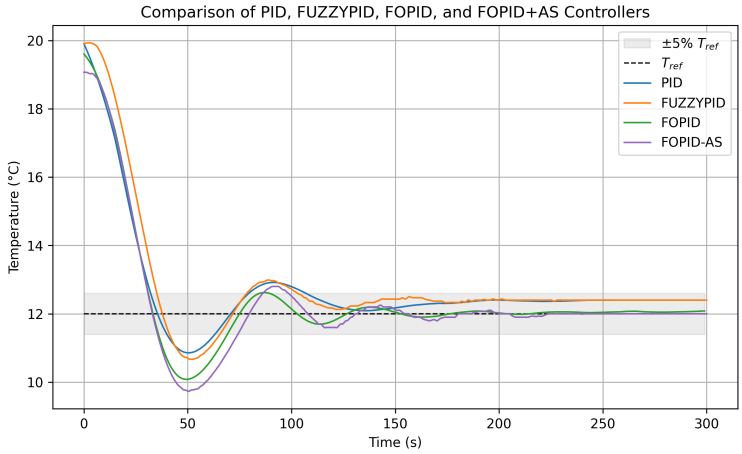

Figure 6. Comparison of the temperature evolution of the Peltier cell under PID, Fuzzy-PID, FOPID, and FOPID-AS controllers. The dotted black line indicates the setpoint T_{ref}, and the shaded gray area represents ±5% tolerance. FOPID and FOPID-AS reach the setpoint, while PID and Fuzzy-PID remain within the tolerance band.

5.5. Peltier Power-Duty Cycle Calibration

To accurately estimate the energy consumption of the Peltier module, a calibration experiment was performed to relate the PWM duty cycle to the actual electrical power. The Peltier was operated at different PWM levels, and the voltage and current were measured to compute the instantaneous power. Table 12 summarizes the experimental results.

Figure 7 shows the resulting relationship between PWM duty cycle and measured Peltier power. A linear regression confirms that the linear approximation used for energy calculation in the controller experiments is accurate, with negligible fitting error. The calibrated curve was then employed to compute the total energy consumed during transient and steady-state operation, replacing the prior assumption based solely on the nominal power.

Table 12. Experimental Peltier power as a function of PWM duty cycle.

Time (s)	PWM	ΔT (°C)	V (V)	I (A)	P (W)
0–60	0	0.0	12	0.00	0.00
60–120	50	4.6	12	0.29	3.48
120–180	150	16.6	12	0.80	9.60
180–240	200	26.5	12	0.93	11.16
240–300	255	33.4	12	1.09	13.08

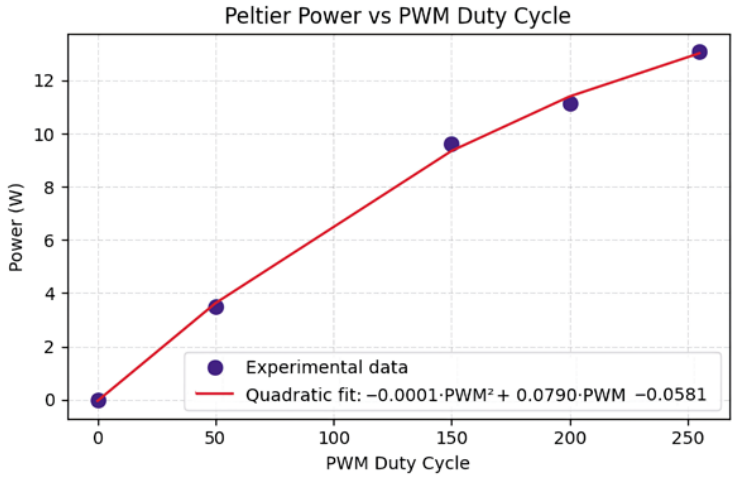

Figure 7. Measured Peltier power as a function of PWM duty cycle. The dashed line represents the linear regression used for energy calculation.

5.6. Comparison of Metrics

The performance metrics are divided into transient and steady state, using $t = 105$ s as the transition point. The settling times obtained for each control strategy under a 5% tolerance band criterion are presented below and summarized in Table 13. These performance metrics were computed from the experimental data and include indicators of transient response (e.g., settling time $t_{\pm 0.5}$, ITAE, and transient energy), as well as steady-state accuracy and stability, expressed by IAE, RMSE, the steady-state deviation (σ_{ss}), and energy consumption. The PID shows the longest settling time ($t_s = 105$ s), defining the boundary between transient and steady state. The FOPID achieves the shortest time ($t_s = 88$ s), while the Fuzzy-PID and FOPID-AS present intermediate values ($t_s = 102.73$ s and $t_s = 98$ s, respectively), with FOPID-AS still offering higher precision and energy efficiency in steady state.

Table 13. Settling times (t_s) with a tolerance band of 5%.

Control	t_s (s)
PID	105.00
FuzzyPID	102.73
FOPID	88.00
FOPID-AS	98.00

Based on the time threshold defined above, Table 14 shows the main performance metrics during the transient phase ($t < 105$ s), including indicators of speed, tracking quality, and energy consumption. The electrical energy consumed by the Peltier module was estimated by integrating the instantaneous power computed from a calibration experiment relating the PWM duty cycle to the actual measured voltage and current. A cubic polynomial fit was applied to the experimental data (Table 12) to interpolate the instantaneous

power for any PWM value. The time integration was performed using the trapezoidal method, and the resulting energy was expressed in Wh, providing an accurate measure of the real energy consumption under the applied control signals.

Table 14. Comparison of transient metrics ($t < 105$ s).

Control	Undershoot (%)	$t_{\pm 0.5}$ (s)	ITAE (°C·s)	Energy (Wh)
PID	9.42	33.00	5148.06	0.18
Fuzzy-PID	11.08	35.51	6102.42	0.20
FOPID	16.08	32.00	5408.98	0.19
FOPID-AS	18.92	31.20	6612.97	0.18

During the transient phase ($t < 105$ s), the FOPID-AS presented the lowest $t_{\pm 0.5}$ (31.20 s) and a relatively high undershoot (18.92%) compared to PID and Fuzzy-PID. Fractional controllers also showed comparable energy consumption to PID. Analysis of Table 14 shows that the FOPID and FOPID-AS controllers achieve faster response times (32.00 s and 31.20 s) but higher undershoot (16.08% and 18.92%). This undershoot arises from excessive initial proportional correction combined with the adaptation of fractional orders, causing the temperature to temporarily drop below the reference. Fine-tuning parameters, such as limiting Kp or adjusting λ and μ, can reduce undershoot without sacrificing response speed. Redefining the bounds of the adaptive supervisor also helps smooth the initial response and prevent excessive negative peaks. Table 15 summarizes accuracy, stability, and energy metrics for $t \geq 105$ s. In steady state, FOPID-AS exhibits the lowest steady-state error e_{ss} = 0.08 °C and energy consumption (0.30 Wh), while its peak-to-peak ripple = 0.67 °C is slightly higher than PID and FOPID. Overall, FOPID-AS maintains high precision and efficiency in steady state, and during the transient it significantly reduces $t_{\pm 0.5}$ compared to PID.

Table 15. Comparison of metrics in steady state ($t \geq 105$ s).

Control	e_{ss} (°C)	σ_{ss} (°C)	Ripple (°C)	Energy (Wh)
PID	0.34	0.11	0.57	0.31
Fuzzy-PID	0.38	0.07	0.40	0.31
FOPID	0.08	0.07	0.53	0.32
FOPID-AS	0.08	0.10	0.67	0.30

Figure 8 shows the evolution of the cumulative absolute error (CAE), where the PID accumulates the largest error, while the fractional schemes, including Fuzzy-PID, FOPID, and FOPID-AS, maintain the lowest slope in steady state ($t \geq 105$ s). This reflects the improvement in efficiency and thermal stability provided by the fractional control strategies and adaptive supervision.

5.7. Comparative Evaluation of Transient and Steady-State Regimes

The performance of the control strategies was evaluated by segmenting the thermal response into two operating phases: the transient regime, characterized by reference tracking dynamics, and the steady-state regime, defined by setpoint stability and accuracy. Table 16 compiles the error indices and temporal parameters derived from both states.

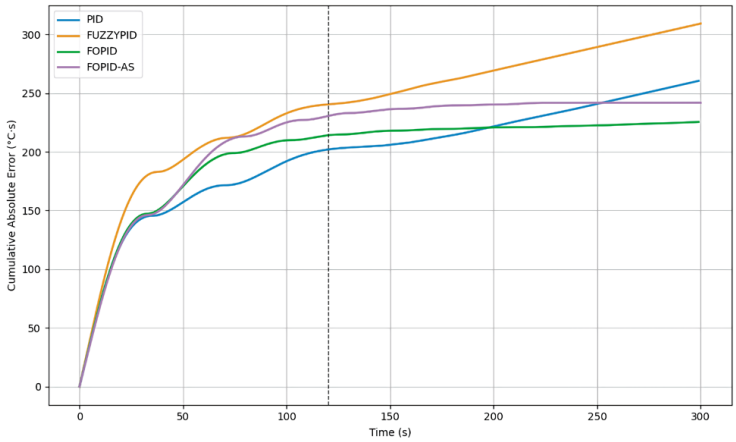

Figure 8. Evolution of the cumulative absolute error (CAE) for PID, FOPID, and FOPID with adaptive supervision. The dashed line indicates the onset of steady-state operation ($t = 105$ s).

Table 16. Dynamic and steady-state performance metrics for the evaluated controllers.

Control	Transient Regime			Steady-State Regime			
	$t_{\pm 0.5}$ (s)	ITAE	Energy (Wh)	e_{ss} (°C)	IAE	RMSE	σ_{ss} (°C)
PID	33.00	5148.06	0.18	0.34	65.00	0.35	0.11
Fuzzy-PID	35.51	6102.42	0.20	0.38	73.24	0.38	0.07
FOPID	32.00	5408.98	0.19	0.08	15.34	0.11	0.11
FOPID-AS	31.20	6612.97	0.18	0.08	14.94	0.13	0.10

In the transient regime ($t < 105$ s), the FOPID-AS controller reaches the band of ± 0.5 °C in $t_{\pm 0.5} = 31.20$ s, compared to 33.00 s for PID, 35.51 s for Fuzzy-PID, and 32.00 s for FOPID. It also achieves one of the lowest ITAE values (6612.97), while maintaining comparable energy consumption (0.18 Wh) to the other controllers.

In steady state ($t \geq 105$ s), FOPID-AS presents a low mean error (0.08 °C) and energy consumption (0.30 Wh), with a peak-to-peak ripple of 0.67 °C. Compared to PID, Fuzzy-PID, and FOPID, FOPID-AS maintains high precision and thermal stability, reflecting a reduced dispersion of the thermal error with respect to the setpoint while still being energy-efficient.

5.8. Thermal Perturbation Experiments

To further evaluate the robustness of the controllers under realistic disturbances, a set of thermal perturbation experiments was conducted. Each controller (PID, Fuzzy-PID, FOPID, and FOPID-AS) was tested at a reference temperature of 12 °C. A hot airflow at 40 °C and 21 km/h was applied from 60 s to 120 s during 600 s trials. Each experiment was repeated three times to ensure reproducibility. Key performance metrics were computed, including the maximum absolute error, integral of absolute error (IAE), mean error and root mean square error (RMSE).

To quantify the dynamic response of each controller to thermal disturbances, the settling time was computed following the maximum of the perturbation. In this study, settling time was defined as the time required for the surface temperature to remain within ± 0.5 °C of the reference setpoint after the disturbance. For PID and Fuzzy-PID, the surface temperature did not fully re-enter the ±0.5 °C band during the trial, resulting in undefined settling times. Table 17 summarizes the performance metrics under thermal perturbation. Figure 9 illustrates the surface temperature responses of all controllers during the perturbation, highlighting the applied disturbance. Figure 10 shows the corresponding control errors over time, while Figure 11 provides the error distribution during the perturbation interval, emphasizing the variability and median behavior of each controller. The results indicate

that the FOPID-AS controller achieves faster stabilization and reduced error magnitude compared to PID and Fuzzy-PID.

Table 17. Performance metrics under thermal perturbation.

Controller	Max Error (°C)	Mean Error (°C)	RMSE (°C)	IAE (°C·s)	Settling Time (s)
PID	8.27	1.95	2.35	1206.70	–
Fuzzy-PID	7.70	1.22	1.72	802.30	–
FOPID	8.83	1.08	1.82	720.12	534.00
FOPID-AS	8.17	0.99	1.71	692.12	494.78

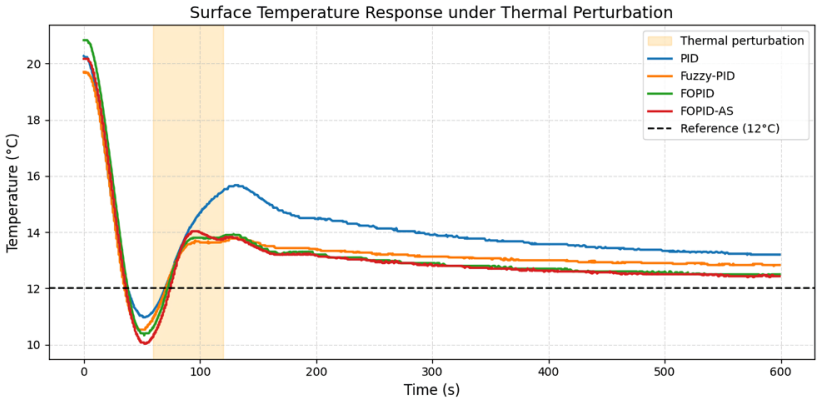

Figure 9. Surface temperature response under thermal perturbation for PID, Fuzzy-PID, FOPID, and FOPID-AS. The shaded area indicates the perturbation interval (60–120 s), and the dashed line represents the reference temperature of 12 °C.

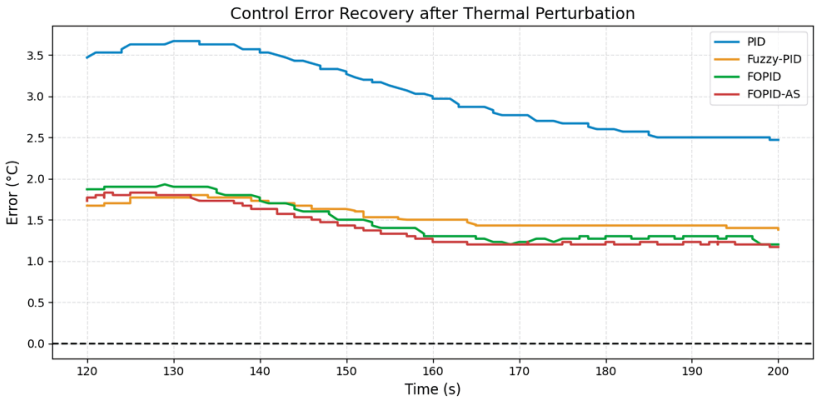

Figure 10. Control error recovery after thermal perturbation for all controllers.

From the metrics, FOPID-AS achieved the lowest mean error (0.99 °C) and post-perturbation standard deviation (0.34 °C), indicating the most precise and consistent temperature control during and after the disturbance. Classical PID showed the highest maximum error (8.27 °C) and IAE (1206.7 °C·s), reflecting slower recovery and larger deviations. Fuzzy-PID reduced the error compared to PID (mean error 1.22 °C), while the standard FOPID controller offered faster recovery (387 s) but slightly higher variability than FOPID-AS. Overall, FOPID-AS provided the best disturbance rejection, combining low error magnitude, fast recovery, and minimal post-perturbation fluctuation, whereas PID was the least robust under the applied thermal perturbation.

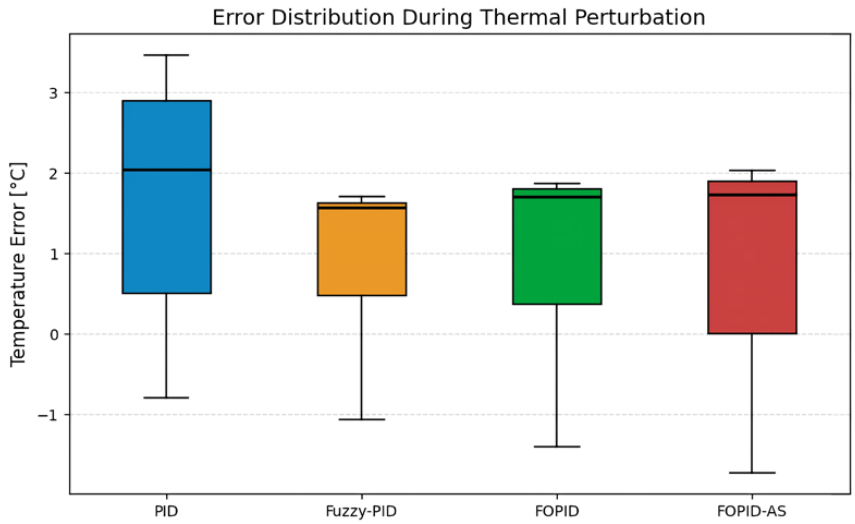

Figure 11. Error distribution during thermal perturbation (60–120 s) for all controllers. The central line shows the median, the box spans the interquartile range, and the whiskers indicate the 5th and 95th percentiles.

5.9. Computational Cost

The FOPID-AS controller exhibits a low computational demand, as summarized in Table 18. The average controller execution time is 0.008 ms, representing a negligible fraction of the total control loop duration. The full control cycle averages 99.35 ms, corresponding to an effective frequency of 10.07 Hz, which aligns with the slow thermal dynamics of Peltier-based systems. CPU and memory usage remain low, at 33.89% and 58.76 MB, respectively.

Table 18. Computational metrics of the FOPID-AS controller during experimental control.

Metric	Value
Mean control loop (ms)	99.35
Mean FOPID-AS time (ms)	0.008
Mean latency (ms)	99.34
Control frequency (Hz)	10.07
CPU usage (%)	33.89
RAM usage (MB)	58.76

During the experimental operation, processor usage averaged 33.89%, and memory consumption stabilized around 58.76 MB, indicating that the FOPID-AS controller can execute in real time on low-cost computing platforms without affecting loop timing. The duration of each control cycle was monitored throughout the experiment. Figure 12 presents the evolution of these cycle times, illustrating the consistency of the control loop across the measurement period.

The inference time of the FOPID-AS controller was recorded during each control cycle, with the results shown in Figure 13, illustrating the evolution of execution times throughout the experimental run. In parallel, processor utilization was monitored to evaluate the computational load imposed by the controller during operation (Figure 14). These measurements indicate consistent execution times and stable processor usage throughout the test, demonstrating predictable computational behavior.

The measured average control loop duration and FOPID-AS inference time are only a small fraction of the characteristic thermal response time of the Peltier system. To evaluate the effect of this latency on system stability, the phase lag at the dominant thermal frequency $f_{\text{dom}} \approx 0.01$ Hz was estimated using $\phi = 360^\circ \cdot f_{\text{dom}} \cdot T_{\text{loop}}$. With an average loop time

of $T_{\text{loop}} \approx 0.099$ s, the resulting phase lag is minimal ($\phi \approx 0.36°$), confirming that the control loop latency does not adversely affect the phase margin or the overall stability of the system. This analysis supports the real-time operation of the controller under the measured experimental conditions.

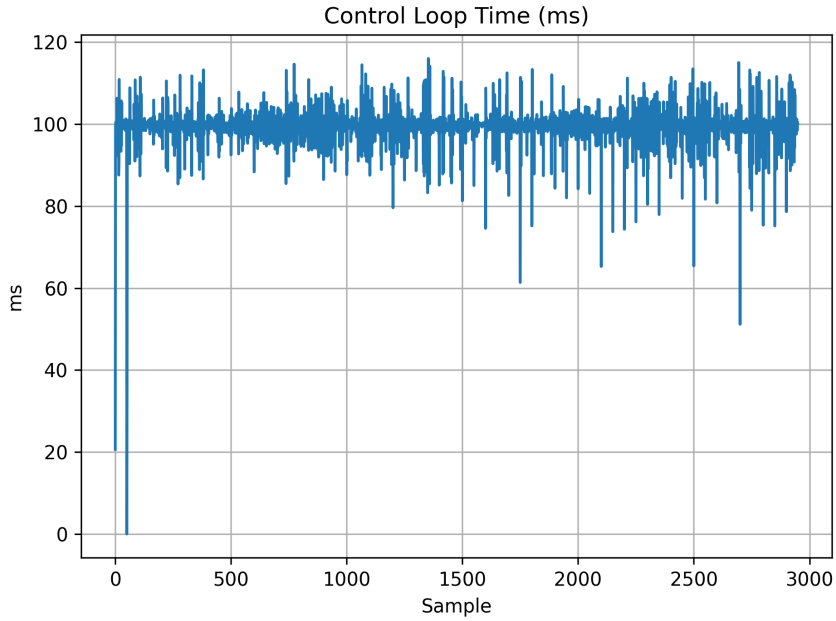

Figure 12. Control cycle time of the FOPID-AS controller during the experimental run. The average loop time remains close to 99.35 ms, with small variations across the test duration.

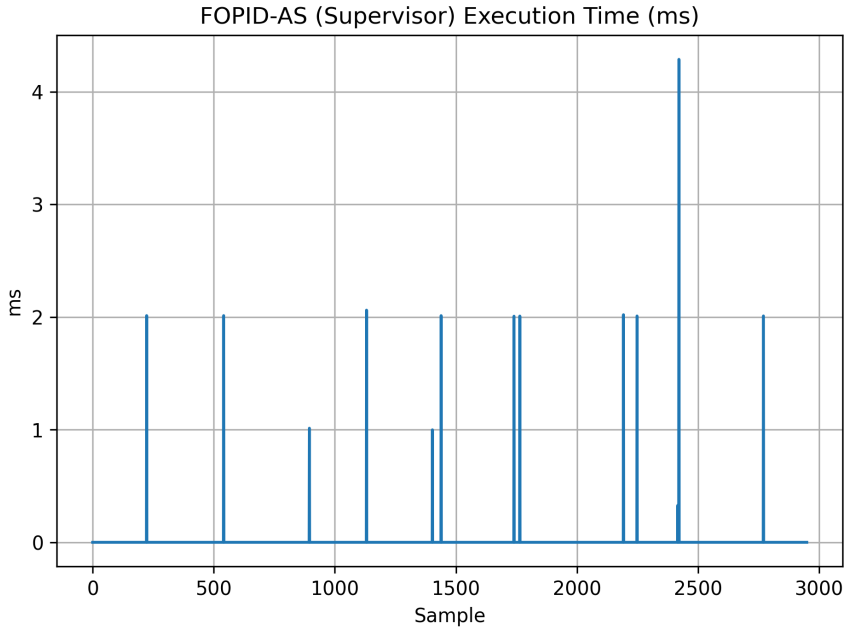

Figure 13. Inference time of the FOPID-AS controller during the experimental run. The average duration remains close to 0.008 ms, with minimal variations throughout the test.

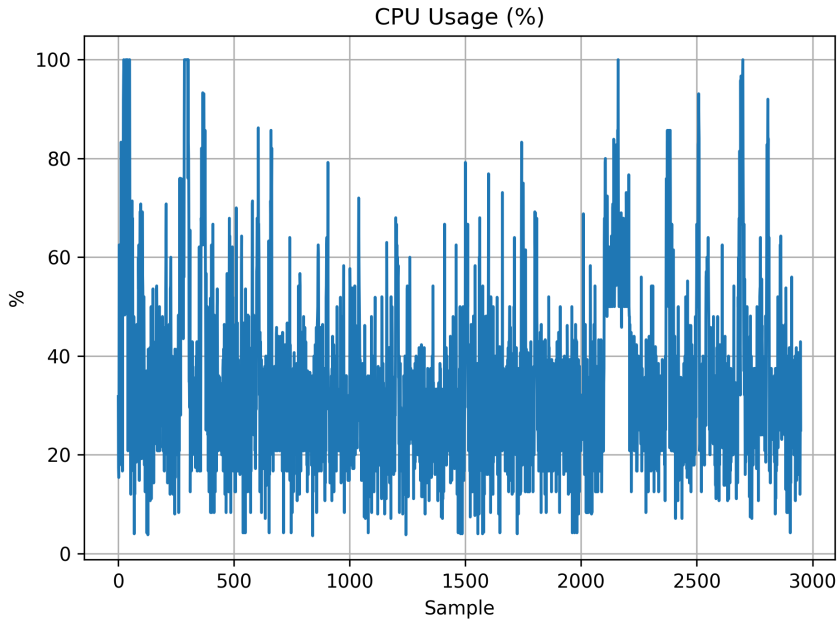

Figure 14. CPU usage during the experimental run with the FOPID-AS controller. The average processor utilization remains around 33.89% with minor fluctuations throughout the test.

6. Discussion

The FOPID-AS hybrid strategy demonstrates faster transient response ($t < 105$ s) compared to PID and FOPID, achieving lower ITAE while maintaining comparable energy consumption. In steady state ($t \geq 105$ s), FOPID-AS exhibits low mean error and steady-state standard deviation σ_{ss}, with a peak-to-peak ripple similar to the other fractional controllers. Quantitative error metrics derived from both conventional statistics and Kernel Density Estimation (standard deviation, kurtosis, and 95% width) confirm that the control error remains more tightly concentrated near zero than in PID and standard FOPID controllers, supporting the claim of sustained accuracy and robustness.

Implementation in embedded systems requires careful consideration of fractional-derivative sensitivity to high-frequency noise [8], which is addressed by dynamically adapting the order μ and applying exponential filtering to the thermistor signal. PWM saturation, potentially inducing nonlinear behavior, is mitigated through smooth ramping and software saturation limits within the Arduino embedded layer [5]. The Grünwald–Letnikov approximation relies on a historical memory length $M = 30$, balancing real-time execution via the Short Memory Principle [13] against the need to capture long-range Peltier dynamics [7]. The distributed architecture among Arduino, Processing, and Python allows supervision without overloading the microcontroller, validating the effectiveness of low-complexity adaptive schemes for nonlinear systems [15].

Although the FOPID-AS controller parameters are initialized offline via PSO tuning, effective adaptation occurs online through the rule-based supervisor, which dynamically adjusts K_p, λ, μ, and K_{ff}. Offline optimization serves only as a starting point and does not constrain the system's ability to respond to changing operating conditions. Experimental metrics show that the average control loop time is approximately 99 ms (≈10 Hz), while the FOPID-AS algorithm itself executes in only ∼0.008 ms, indicating that most latency arises from communication overhead. This latency has negligible impact on control performance due to the slow thermal dynamics of the Peltier system. Consequently, neither communication delays nor offline tuning constitute significant limitations for stability or control effectiveness under the evaluated conditions. To quantitatively assess the supervisor's effectiveness, the transient response was divided into three phases (aggressive, transition,

fine) according to the magnitude of the absolute control error. Table 19 summarizes the overshoot, time within ±0.5 °C, and mean adaptive parameters for each phase.

Table 19. FOPID-AS transient response metrics per phase.

Phase	Overshoot (%)	Time in ±0.5 °C (s)	Kp Mean	λ Mean	μ Mean
aggressive	58.33	0.0	76.609	0.550	1.100
transition	25.00	41.3	63.910	0.756	1.614
fine	1.67	268.3	47.144	0.850	1.850

These results illustrate that the adaptive supervisor effectively reduces overshoot and fine-tunes the response as the error decreases, confirming the benefits of phase-dependent parameter adjustment.

7. Conclusions

This work evaluated the integration of a rule-based bounded gain-scheduling supervisor with a PSO-optimized fractional-order PID (FOPID) controller in a thermoelectric system. Experimental results demonstrate that the FOPID-AS controller improves both transient and steady-state performance. In the transient phase, it achieves lower ITAE and IAE values than PID, Fuzzy-PID, and static FOPID controllers, reaching the ±0.5 °C tolerance band more rapidly. Steady-state performance shows reduced mean error and standard deviation, while maintaining comparable energy consumption and peak-to-peak ripple.

The adaptive supervisor dynamically adjusts multiple controller parameters (K_p, λ, μ, and K_{ff}) based on instantaneous error, enabling rapid compensation for disturbances while preserving overall stability. Parameter variations are closely correlated with system deviations, ensuring effective error confinement. Statistical analysis of the error distribution (standard deviation, kurtosis, and 95% width) further confirms that the FOPID-AS keeps the control error tightly centered around the setpoint, reinforcing the improvements observed in both transient and steady-state metrics.

The distributed architecture across Arduino, Python supervision, and visualization layers allows real-time execution without introducing performance limitations from communication latency or computation overhead. The combination of fractional-order memory effects and adaptive parameter scheduling enables accurate capture of thermal dynamics while preserving computational efficiency.

Overall, the results provide a quantitative characterization of the FOPID-AS controller, demonstrating improvements in accuracy, robustness, and efficiency relative to conventional PID and FOPID approaches. Future work will explore self-learning fractional control architectures with real-time adaptive supervision using neural networks to address highly dynamic conditions and unmodeled disturbances.

Author Contributions: Conceptualization, M.F.F.P., C.S.M., F.L.Z. and A.R.M.; methodology, M.F.F.P. and A.R.M.; software, M.F.F.P. and C.S.M.; validation, M.F.F.P., F.L.Z. and A.R.M.; formal analysis, M.F.F.P., F.L.Z. and A.R.M.; investigation, M.F.F.P., C.S.M. and A.R.M.; resources, F.L.Z. and A.R.M.; data curation, M.F.F.P.; writing—original draft preparation, M.F.F.P., C.S.M. and A.R.M.; writing—review and editing, M.F.F.P., C.S.M., F.L.Z. and A.R.M.; visualization, M.F.F.P.; supervision, C.S.M., F.L.Z. and A.R.M.; project administration, C.S.M., F.L.Z. and A.R.M.; funding acquisition, C.S.M., F.L.Z. and A.R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-funded by the INTERREG V-A Cooperation Spain–Portugal MAC (Madeira-Azores-Canarias) program via the MITIMAC project MAC2/1.1a/263. This research was co-funded by Project PIE 2023-60 CONVOCATORIA DE PROYECTOS DE INNOVACIÓN EDUCATIVA

2023 de la Universidad de Las Palmas de Gran Canaria. This research was co-funded by Project PIE-INT 2024-29 CONVOCATORIA DE PROYECTOS DE INNOVACIÓN EDUCATIVA 2024 de la Universidad de Las Palmas de Gran Canaria.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to acknowledge the availability of the technical resources in the laboratories of the University of Las Palmas de Gran Canaria, which were essential for the design and development of the instruments and devices necessary for the completion of this research project.

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

Abbreviations

The following abbreviations are used in this manuscript:

CAE	Cumulative Absolute Error
FOPID	Fractional-Order Proportional–Integral–Derivative
FOPID-AS	Fractional-Order Proportional–Integral–Derivative with Adaptive Supervision
GL	Grünwald–Letnikov
IAE	Integral Absolute Error
ISE	Integral Square Error
ITAE	Integral Time Absolute Error
KDE	Kernel Density Estimation
NTC	Negative Temperature Coefficient
PID	Proportional–Integral–Derivative
PSO	Particle Swarm Optimization
PWM	Pulse Width Modulation
RH	Relative Humidity
RMSE	Root Mean Square Error
TCP	Transmission Control Protocol

References

1. Åström, K.J.; Murray, R.M. *Feedback Systems: An Introduction for Scientists and Engineers*; Princeton University Press: Princeton, NJ, USA, 2008; pp. 301–309.
2. Tavazoei, M.S.; Tavakoli-Kakhki, M. Compensation by fractional-order phase-lead/lag compensators. *IET Control Theory Appl.* **2014**, *8*, 319–329. [CrossRef]
3. Han, J. From PID to Active Disturbance Rejection Control. *IEEE Trans. Ind. Electron.* **2009**, *56*, 900–906. [CrossRef]
4. Padula, F.; Visioli, A. On the fragility of fractional-order PID controllers for FOPDT processes. *ISA Trans.* **2016**, *60*, 228–243. [CrossRef] [PubMed]
5. Marwedel, P. *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things*, 4th ed.; Springer Nature: Cham, Switzerland, 2021; pp. 138–140.
6. Naifar, O.; Ben Makhlouf, A. (Eds.) *Fractional Order Systems—Control Theory and Applications: Fundamentals and Applications*; Springer Nature: Cham, Switzerland, 2022; pp. 143–144.
7. Petráš, I. *Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation*; Springer: Berlin/Heidelberg, Germany; Higher Education Press: Beijing, China, 2011; p. 47.
8. Bashishtha, T.K.; Singh, V.P.; Yadav, U.K.; Sahu, U.K. Fractional-order PID controllers and applications: A comprehensive survey. *Annu. Rev. Control* **2025**, *60*, 101013. [CrossRef]
9. Podlubny, I. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*; Academic Press: San Diego, CA, USA, 1999; Volume 198.
10. Podlubny, I. *Geometric and Physical Interpretation of Fractional Integration and Fractional Differentiation*; Technical Report TUKE–10–2001; Technical University of Kosice: Kosice, Slovakia, 2001.
11. Padula, F.; Visioli, A. Tuning rules for optimal PID and fractional-order PID controllers. *J. Process Control* **2011**, *21*, 69–81. [CrossRef]
12. Monje, C.A.; Vinagre, B.M.; Feliu, V.; Chen, Y. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng. Pract.* **2008**, *16*, 798–812. [CrossRef]

13. Monje, C.A.; Chen, Y.; Vinagre, B.M.; Xue, D.; Feliu, V. *Fractional-Order Systems and Controls: Fundamentals and Applications*; Springer: London, UK, 2010; p. 323.
14. Ladaci, S.; Loiseau, J.J.; Charef, A. Fractional order adaptive high-gain controllers for a class of linear systems. *Commun. Nonlinear Sci. Numer. Simul.* **2008**, *13*, 707–714. [CrossRef]
15. Yin, C.; Cheng, Y.; Chen, Y.; Stark, B.; Zhong, S. Adaptive fractional-order switching-type control method design for 3D fractional-order nonlinear systems. *Nonlinear Dyn.* **2015**, *82*, 39–52. [CrossRef]
16. Morris, A.S.; Langari, R. *Measurement and Instrumentation: Theory and Application*; Academic Press: Waltham, MA, USA, 2012; p. 364.
17. Tepljakov, A.; Alagoz, B.B.; Yeroglu, C.; Gonzalez, E.; HosseinNia, S.H.; Petlenkov, E. FOPID controllers and their industrial applications: A survey of recent results. In *Proceedings of the IFAC-PapersOnLine*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 51, pp. 25–30.
18. Petráš, I. Tuning and Implementation Methods for Fractional-Order Controllers. *Fract. Calc. Appl. Anal.* **2012**, *15*, 282–303. [CrossRef]
19. Zamani, A.; Barakati, S.M.; Yousofi-Darmian, S. Design of a fractional order PID controller using GBMO algorithm for load–frequency control with governor saturation consideration. *ISA Trans.* **2016**, *64*, 56–66. [CrossRef] [PubMed]
20. Tepljakov, A.; Alagoz, B.B.; Yeroglu, C.; Gonzalez, E.A.; HosseinNia, S.H.; Petlenkov, E.; Ates, A.; Cech, M. Towards Industrialization of FOPID Controllers: A Survey on Milestones of Fractional-Order Control and Pathways for Future Developments. *IEEE Access* **2021**, *9*, 21016–21042. [CrossRef]