Journal of Artificial Intelligence and Technology, (Ahead of Print)
https://doi.org/10.37965/jait.2026.1328

ISTP
RESEARCH ARTICLE

Ensemble-Based PID Gain Learning Framework for Adaptive Temperature Control in Micro-Thermoelectric Systems

Mandhapalli Anitha and Sankata Bhanjan Prusty
School of Electronics and Communication Engineering, REVA University, Bengaluru, India

(Received 10 April 2026; Revised 08 July 2026; Accepted 16 July 2026; Published online 15 August 2026)

Abstract: Precise temperature regulation in Micro-Thermoelectric Cooler (Micro-TEC) systems is essential for applications such as polymerase chain reaction (PCR) thermal cyclers, biomedical instrumentation, and microelectronic cooling platforms. Proportional-Integral-Derivative (PID) controllers remain widely adopted for their structural simplicity, but their performance depends heavily on gain selection. Heuristic tuning has limited adaptability under nonlinear thermal dynamics, while optimization-based techniques such as genetic algorithm (GA) and particle swarm optimization (PSO) require repeated, computationally intensive searches. A consolidated PID gain knowledge repository is built by running GA and PSO offline over 2,500 Micro-TEC operating scenarios, which serves as supervised training data for an ensemble regression model composed of Random Forest, Extra Trees, and Gradient Boosting learners combined through stacking. The trained ensemble model learns the mapping between operating conditions and near-optimal PID gains, eliminating the need for repeated evolutionary optimization during deployment. On the Micro-TEC plant model, the proposed ensemble controller lowers overshoot by 37.9% (from 6.31% to 3.92%), cuts settling time by 28.0% (from 16.8 s to 12.1 s), reduces steady-state temperature error to 0.11°C (a 59.3% reduction), and selects its gains about 65 times faster (15 ms against 980 ms) than a conventional PSO-tuned PID. The results show that the framework bridges classical PID control and data-driven learning, providing a scalable, computationally efficient approach for scenario-aware gain scheduling in nonlinear thermal systems.

Keywords: ensemble learning; genetic algorithm; micro-thermoelectric cooling; particle swarm optimization; PID controller

I. INTRODUCTION

Proportional-Integral-Derivative (PID) controllers continue to be widely employed in industrial and embedded control applications owing to their simple structure, ease of implementation, and robustness across a broad range of systems. The practical effectiveness of a PID controller, however, is fundamentally determined by the appropriate selection of its tuning parameters, the proportional, integral, and derivative gains. Improperly tuned gains can lead to excessive overshoot, sluggish response, oscillatory behavior, or instability, particularly in systems with nonlinear dynamics and operating-condition variability [1].

Traditional PID tuning approaches, such as rule-based methods and classical heuristics, offer limited adaptability and are often designed around nominal operating points. While optimization-based tuning techniques improve performance by systematically searching the parameter space, they typically rely on repeated optimization runs and are not well suited for scalable or data-intensive scenarios. These limitations have motivated increasing interest in data-driven PID tuning strategies, where learning algorithms are employed to capture the relationship between system behavior and effective controller parameters. Within this context, PID tuning can be formulated as a supervised regression problem, where the objective is to predict suitable controller gains based on observed system characteristics. Rather than modifying the control law itself, learning-based approaches retain the classical PID structure and focus exclusively on automating the gain selection process. This formulation preserves the transparency and low computational cost of PID control while enabling improved performance across varying operating regimes [2,3].

Ensemble learning techniques have emerged as a particularly effective class of models for data-driven PID tuning. By aggregating the predictions of multiple weak learners, ensemble methods are capable of approximating complex, nonlinear mappings between system states and controller parameters while maintaining robustness to noise and model uncertainty. In PID tuning applications, ensemble models are trained to estimate continuous-valued gain parameters from features that describe the current control condition, such as tracking error characteristics or system response indicators. A key distinction of ensemble learning-based PID tuning lies in its operational role: the learning model functions as a gain prediction mechanism rather than a real-time optimizer or controller replacement. Once trained, the ensemble model provides immediate estimates of appropriate PID parameters, which are then applied within a conventional PID controller. Performance evaluation is typically conducted using standard control metrics, overshoot, settling time, steady-state error, and stability, allowing direct comparison with classical and optimization-based tuning methods. As a result, ensemble learning offers a scalable and systematic pathway for improving PID controller performance while preserving the structural simplicity that underpins PID's continued relevance in practical control systems [4–8]. Micro-Thermoelectric Cooler (Micro-TEC) systems are widely used in precision temperature regulation applications where accurate and stable thermal control is critical. Although PID controllers are commonly employed due to their simplicity and ease of implementation, their performance is highly sensitive to gain selection. In nonlinear thermal systems, variations in ambient temperature,

Corresponding author: Mandhapalli Anitha (e-mail: anitham_12@rediffmail.com).

load conditions, and plant parameters can significantly degrade tracking accuracy when fixed or heuristically tuned gains are used.

Evolutionary optimization techniques such as genetic algorithm (GA) and particle swarm optimization (PSO) have been applied to improve PID tuning performance. However, these methods require repeated computationally intensive searches whenever operating conditions change, limiting their practicality for adaptive or real-time applications [9,10].

A. RESEARCH STATUS OF ADAPTIVE TEMPERATURE CONTROL IN MICRO-TEC SYSTEMS

Micro-TECs are solid-state Peltier devices that provide compact, vibration-free, and bidirectional thermal control. They are used in polymerase chain reaction (PCR) thermal cyclers, biomedical analyzers, laser diode and photonic stabilization, and the localized cooling of microelectronics. The thermal response of a Micro-TEC is nonlinear. The Peltier heat, the Joule self-heating, and the conductive exchange with the hot side together link the drive current to the temperature. This coupling changes with the operating point, with the heat-sink and ambient conditions, and with the thermal load. For this reason, fixed or heuristically tuned PID gains that work well at one setpoint often perform poorly at another.

Reported methods for Micro-TEC temperature control include fixed-gain PID with anti-windup, gain scheduling, fuzzy and adaptive PID, and model predictive control (MPC). More recent work applies machine learning to PID design. Representative examples are the learning-based controller reported for the Micro-TEC of a PCR device [6] and the machine learning performance assessment of PID loops [7]. The present work differs in that it does not retune the controller online. The improved GA and PSO are run once, offline, over a fixed set of operating scenarios. The gains they produce are stored and used to train a single ensemble model. This model then returns suitable gains for new operating conditions without any further search.

B. MOTIVATION AND CONTRIBUTION OF RESEARCH WORK

Despite their long-standing adoption, PID controllers remain widely used in modern control systems because of their simplicity, interpretability, and compatibility with resource-constrained platforms. As control applications increasingly involve nonlinear dynamics and variable operating conditions, the demand for PID tuning strategies that go beyond fixed heuristics has become more pronounced. In practice, maintaining consistent control performance across different regimes often requires repeated manual retuning or reliance on computationally intensive optimization procedures, both of which limit scalability and practical deployment. Recent advances in machine learning have created new opportunities for enhancing classical control frameworks without abandoning their structural advantages. In particular, learning-based approaches enable the extraction of patterns from historical or synthesized data, allowing controller parameters to be selected in a systematic and automated manner. When applied to PID tuning, such approaches offer the potential to reduce dependence on trial-and-error methods while improving consistency in performance metrics such as overshoot, response time, and stability. Among data-driven techniques, ensemble learning has demonstrated strong generalization capability in regression problems involving nonlinear relationships and uncertainty. Its ability to aggregate multiple learners makes it well suited for predicting continuous-valued controller gains under varying conditions while remaining computationally efficient once trained. This characteristic is especially attractive for PID tuning scenarios, where rapid gain selection is required but online optimization is impractical. The motivation for this work therefore arises from the need to bridge classical PID control and modern data-driven methods. Ensemble learning is used to automate PID parameter selection. This keeps the simplicity, transparency, and reliability that underpin the widespread use of PID controllers in practical control systems.

1). MICRO-TEC-ORIENTED OPTIMIZATION FRAMEWORK. A structured offline optimization strategy is developed specifically for nonlinear Micro-TEC temperature regulation systems. Improved GA and PSO procedures are employed across varying operating conditions to generate high-quality PID gain configurations.

Unified PID Gain Knowledge Repository Construction: A consolidated PID gain knowledge repository is established by aggregating optimized gain sets obtained from multiple evolutionary optimization runs under diverse thermal scenarios. This repository forms a structured dataset that captures the relationship between plant conditions and effective controller parameters.

Ensemble-Based PID Gain Prediction Model: The PID tuning problem is reformulated as a supervised learning task, enabling direct gain prediction without repeated evolutionary search. An ensemble regression framework composed of Random Forest, Extra Trees, and Gradient Boosting models is integrated using a stacking meta-learner to enhance prediction robustness and generalization capability.

II. RELATED WORK

PID controllers have long been a fundamental component of Nonlinear Control Systems owing to their structural simplicity, robustness, and ease of practical implementation. By appropriately tuning the proportional (P), integral (I), and derivative (D) gains, PID controllers regulate system responses to deviations from desired trajectories, thereby ensuring stable motion control [11]. Despite their widespread adoption, conventional PID controllers exhibit notable shortcomings when applied to complex and dynamic operational environments. One of the primary limitations lies in static parameter tuning. Classical tuning techniques, such as Ziegler–Nichols and Cohen–Coon methods, rely heavily on heuristic rules or trial-and-error procedures. While these approaches can yield acceptable baseline performance, they often fail to achieve optimal control in systems characterized by varying dynamics or uncertainties [12]. PID controllers are also sensitive to environmental and operational changes. Parameters tuned for a specific path geometry, payload, or operating condition may degrade significantly under variations in load, speed, or external disturbances [13]. This lack of adaptability necessitates frequent manual recalibration, which is both time-consuming and impractical in real-world automated guided vehicle (AGV) deployments. Another critical drawback of traditional PID controllers is their limited capability in handling nonlinear dynamics. Many industrial and robotic systems exhibit nonlinear behavior, particularly during high-speed operations, sharp turns, or abrupt maneuvers. In such scenarios, conventional PID controllers often result in overshoot, oscillatory responses, or prolonged settling times, thereby compromising control accuracy and system stability [14]. These challenges highlight the need for advanced tuning strategies and alternative control frameworks capable of dynamic adaptation and robust performance under uncertainty [15]. To overcome these limitations, several optimization-based PID tuning methods have been proposed in the literature. Metaheuristic algorithms such as GA, PSO, and beetle antennae search (BAS) have gained significant attention for automating the PID tuning process [16]. These approaches enable adaptive adjustment of controller parameters based on system feedback, thereby enhancing performance in dynamic environments [17]. Among these methods, PSO has been shown to offer faster convergence and better suitability for real-time applications compared to GA, making it particularly attractive for AGV control systems. Beyond PID optimization, several advanced control strategies have also been explored. Sliding mode control (SMC) provides strong robustness against disturbances and model uncertainties, while MPC offers the ability to anticipate future system behavior and optimize control actions accordingly [18]. These methods have demonstrated promising performance in AGV navigation tasks involving complex trajectories and constrained environments [19]. Fault-tolerant control has emerged as another critical research area, particularly in multi-agent robotic systems such as multi-unmanned aerial vehicle (multi-UAV) formations. Actuator faults, input saturation, and external disturbances can severely affect system stability and coordination. A distributed fault-tolerant control framework employing disturbance observers (DOs) was introduced in [20], enabling real-time estimation of actuator failures and compensation through auxiliary dynamic systems. Similarly, a finite-time containment control strategy was proposed in [21], ensuring formation maintenance despite actuator malfunctions and communication uncertainties. Recent studies further confirm the effectiveness of PSO-based PID optimization for both linear and nonlinear systems. An improved PSO algorithm with adaptive inertia weight was presented in [22], achieving superior control performance in terms of reduced settling time, overshoot, and steady-state error. In [23], a hybrid PSO-Gray Wolf Optimizer (GWO) approach was proposed for nonlinear PID tuning, demonstrating enhanced robustness and faster convergence compared to conventional PSO and GA techniques. A multi-objective PSO (MOPSO) framework was also employed in [24] to optimize PID controllers for quadrotor UAVs, effectively balancing competing objectives such as stability, response speed, and energy efficiency. Reference [25] introduces a stochastic gradient descent (SGD)-based linear function approximation (LFA) framework and extends it by integrating PID control concepts into the learning process. In this work, six distinct PID-enhanced LFA variants are developed, including the standard PID, integral-separated PID, gearshift integral PID, dead-zone PID, anti-windup PID, and incomplete differential PID configurations. These variants are designed to refine instantaneous learning errors within the stochastic gradient update by incorporating historical error information, thereby improving convergence stability and learning efficiency.

III. PROPOSED METHODOLOGY

Control systems are essential in Micro-TEC applications, where precise temperature regulation is essential for systems such as PCR thermal cyclers, biomedical devices, laser diodes, and microelectronic cooling platforms. Due to the fast thermal dynamics and nonlinear characteristics of thermoelectric modules, maintaining accurate and stable temperature control is a significant challenge. The temperature evolution of a Micro-TEC system can be modeled using a lumped thermal capacitance framework. The governing dynamic equation is expressed in eq (1):

$$C\frac{dT(t)}{dt} = R_{TEC}(temp) - R_{loss}(emp) \tag{1}$$

A. PID CONTROL LAW AND PROBLEM FORMULATION

Traditionally, PID controllers are widely employed in Micro-TEC systems because of their simplicity and ease of implementation. The PID control law in eq. (2) and eq. (3) are given as:

$$h(R^{(k)}) \approx h(R^{(k)}) = \int_0^V g(v)^2 dv \tag{2}$$

$$v(u) = L_q f(u) + L_j \int f(u) + L_e \frac{df(u)}{dt} \tag{3}$$

The PID control law regulates the current supplied to the TEC module, considering the temperature error between the desired setpoint and the output. The classical tuning model mainly struggles with issues like overshoot, thermal lag, parameter drift, and nonlinear heat transfer. Recent research has focused on optimization through PSO- and GA-based optimization to minimize steady-state error and thermal settling and state error rate.

Figure 1 shows the resulting closed-loop control architecture, in which the ensemble-predicted gains (Kp, Ki, Kd) drive the PID controller regulating the Micro-TEC plant of Eq. (1) against thermal-load and ambient-temperature disturbances.

B. OPTIMIZATION-BASED PID TUNING

The research proposes ensemble learning inside the workflow that tunes PID controllers. A GA and a particle swarm routine still run, but the added software learns from every completed tuning job and reuses that knowledge. As a result, the next tuning converges faster, keeps the loop stable under wider plant conditions, and needs fewer objective function calls. A plain GA evolves strings of three numbers (the PID gains) by breeding, mutating, and simulating each child. One simulation per child is slow. The study trains a network on all earlier simulations. The network then acts as a surrogate. It reads a set of gains and immediately outputs an estimated closed-loop score. Only the most promising children go to the full simulator. The surrogate therefore trims computation time. When the setpoint temperature or the thermal load shifts, a classical GA would discard the old population as well as restart. The machine learning layer avoids this restart. It retains the mapping between past operating points and the gain sets that worked. The search begins inside a smaller, better-informed region. The layer also watches population diversity or short-term history and then raises or lowers mutation probability and crossover probability on the fly. The code keeps the genetic module and adds the surrogate next to shorten each new tuning campaign without sacrificing robustness. The machine learning (ML) model $h(R)$ is trained to get the approximate true fitness function, denoted as $h(R)$, where $R^{(k)} = (M_r^{(k)}, M_k^{(k)}, M_f^{(k)})$. The optimization is shown in eq. (4):

$$R_{new}^{(k)} = GenAlg(R_{old}^{(k)}, h(R^{(k)})) \tag{4}$$

The GA replaces the usual full simulation step with a value called h that a machine learning model provides, whereas this value further guides the search. Machine learning also improves PSO. A surrogate model estimates how well each set of PID gains will perform. The swarm does not need a complete simulation for every candidate. Because the model gives fast feedback, the swarm keeps moving and does not settle too early, and the optimizer shortens the overall run time.

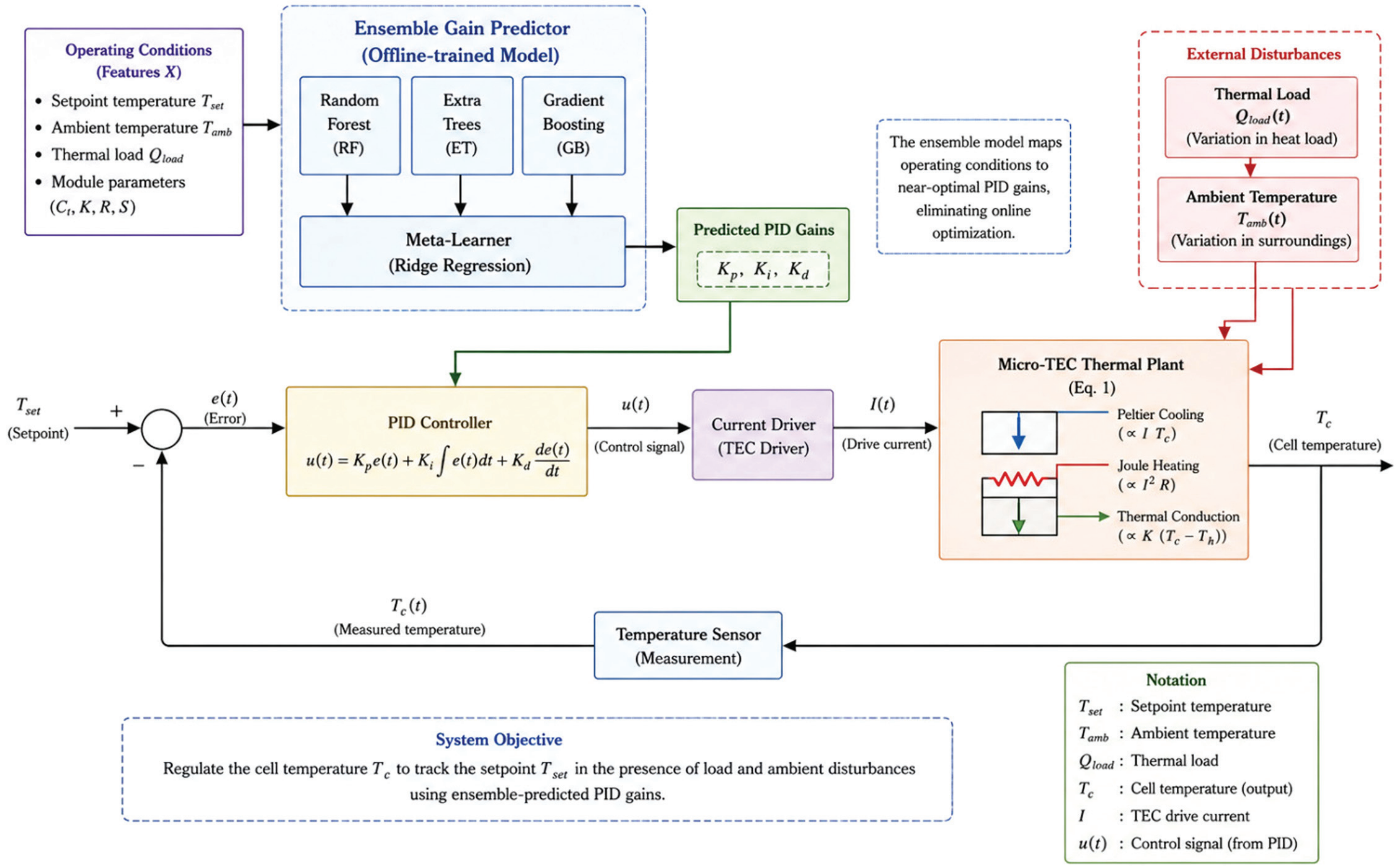

Fig. 1. Control system.

Standard PSO revises every particle velocity with two constant factors: one for the particle's own memory and the other for the group influence. Different parts of the search space, however, demand different amounts of exploration. A small neural network reads the present state of the particle and of the whole swarm and then outputs new values for those two factors on every step. The velocity update rule becomes eq. (5). The eq. (6) is further modified as:

$$x_k(v+1) = yx_k(v) + e_1 t_1 (R_{best} - M_k(v)) + e_2 t_2 (i_{best} - M_k(v)) \quad (5)$$

$$x_k(v+1) = yx_k(v) + e_1(v) t_1 (R_{best} - M_k(v)) + e_2(v) t_2 (i_{best} - M_k(v)) \quad (6)$$

In eq. (6), $e_1(v)$ and $e_2(v)$ denote the acceleration coefficients that a neural network predicts at each iteration v from the current state of the particle and the swarm, replacing the fixed constants used in standard PSO. A neural network learns the patterns and adapts to the swarm's behavior. The machine learning model that guides each particle takes the particle's present position as its starting input $Z_k(v) = (M_r, M_k, M_f)$ to accommodate the velocity $x_k(v+1)$ and improve the fitness function's negative eq. (7):

$$t(v) = -h(Z_k(v)) = -\int_0^V g(v)^2 dv \quad (7)$$

The ML system also learns how to pick gain values for a schedule. The neural network is trained so that it links each set of operating conditions to a set of gains. To build the training data, they run GA or PSO many times, while they vary the plant conditions and store the gains that each run returns. Once training ends, the ML model receives new conditions and outputs the gains that suit them as shown in eq. (8):

$$[M_r, M_k, M_f] = ml\,model(v, E_C) \quad (8)$$

The ML model is a neural network, a decision tree, or another regression method that links the current state of the plant to the best PID gains. When ML joins GA besides PSO, the system adapts faster and needs fewer calculations while it tracks the nonlinear behavior. A neural ML routine searches for the PID set (M_r, M_k, M_f) that still works best when driven to its limits. Those limits appear when the plant parameters, the ambient temperature, or the thermal load reach values that push the Micro-TEC outside the range where normal controllers keep it steady. It includes the reaction rate constant m_0, activation energy G_c, heat transfer coefficient W, and inflow concentration E_C^0; further, the optimization is structured as eq. (9):

$$\min_{M_r, M_k, M_f} \max_{\varnothing} L(M_r, M_k, M_f, \varnothing_c) \quad (9)$$

Here, $L(M_r, M_k, M_f, \varnothing_c)$ defines the cost function based on the error metric. The main aim is to find the PID parameters that minimize the error while considering the parameters $\varnothing_c$, accommodating m_0, G_C, W, which could make the system difficult to control.

The study uses neural networks to improve two tasks: setting PID gains and finding the parameter values that cause the largest control error. Two separate neural models carry out the work. The first model predicts the parameters that create the worst-case behavior. It learns to output the parameter set that produces the greatest tracking error for a fixed set of PID gains. Training this model supplies a fast way to estimate which settings yield the hardest plant to regulate. In the GA optimization, every individual is a single vector that holds both the PID gains in eq. (10):

$$R^{(k)} = (M_r^{(k)},M_k^{(k)},M_f^{(k)},\varnothing_C^{(k)}) \tag{10}$$

The study uses neural networks to improve two tasks: setting PID gains and the parameter values that cause the largest control error. Two separate neural models carry out the work. The first model predicts the parameters that create the worst-case behavior. It learns to output the parameter set that produces the greatest tracking error for a fixed set of PID gains. Training this model supplies a fast way to estimate which settings yield the hardest plant to regulate. In the GA optimization, every individual is a single vector that holds both the PID gains in eq. (11):

$$h(R^{(k)}) = L(M_r^{(k)},M_k^{(k)},M_f^{(k)},\varnothing_C^{(k)}) \tag{11}$$

This cuts the number of full simulations required for every candidate. The GA updates the PID gains. Mutation and selection drive down the error that the neural network forecasts for the most extreme values. In the particle swarm, every particle holds the PID gains together in eq. (12):

$$Z^{(k)} = (M_r^{(k)},M_k^{(k)},M_f^{(k)},\varnothing_C^{(k)}) \tag{12}$$

The velocity and position of each particle change according to the predictions of the neural network outputs. The swarm reaches good PID settings. Neural networks learn from records of many earlier runs. In each run, either a GA or particle swarm algorithm tunes the PID settings at different operating points, (V,E_C^0,m_0,G_c), with the corresponding optimal gains M_r,M_k,M_f. Every record lists an input side and an output side. The input side holds the operating conditions. The neural network mapping is shown in eq. (13):

$$h(Z^{(k)}) = L(M_r,M_k,M_f,\varnothing_c) \tag{13}$$

The trained model takes the parameters and returns the best PID gains, or it takes the gains and returns the parameters. After training, it captures how the plant conditions map to good gains, and the optimizer covers the search space quickly. It also exposes situations that drive the PID to the edge of stability. Adversarial machine learning is used to create those extreme cases on demand. A neural network is taught to pick the set of plant numbers that gives the poorest closed-loop response for a fixed PID. Those hostile cases are injected into the tuning run, forcing the algorithm to pick gains that still work when the plant is at its nastiest. Because the predictor is neural, it remembers earlier runs and needs fewer plant simulations next time.

This cuts the CPU budget, and in the adversarial loop, the task is to choose M_r,M_k,M_f so that the controller survives the worst plant G_c that the inner loop can invent. The inner loop chooses reaction rate constant m_0, activation energy G_c, heat transfer coefficient W, and related terms to maximize control error.

The outer loop adjusts the same three PID numbers to drive that error back down. Equation (2) states the problem: maximize error inside and minimize it outside. Two optimizers run side by side. One GA or Particle Swarm search hunts for the plant the other searches for the PID.

In the GA version, each chromosome is one vector that holds both the PID triplet and the plant parameters, exactly as written in eq. (14). The outer GA minimizes error, while the inner GA maximizes it. The single fitness score tells how well the controller tracks and how the plant acts.

$$[M_r,M_k,M_f] = mod(V,E_C^0,m_0,G_c) \tag{14}$$

The algorithm treats the PID gains and the parameters as two separate groups of genes. It uses crossover, mutation, and selection on both groups in the same cycle. The PID gains change to lower the error signal. These parameters are changed to push the plant dynamics toward conditions that are hard for any controller to handle, as shown in eq. (15):

$$h(R^{(k)}) = \min_{M_r^{(k)},M_k^{(k)},M_f^{(k)}} \max_{\varnothing_C^{(k)}} \int_0^V g(v)^2 dv \tag{15}$$

The evolutionary process adjusts both PID gains besides parameters through crossover, mutation, and selection steps that move toward an optimal outcome. PID gains change, so the error drops. The parameters change, so the system becomes harder to control. Each particle in the PSO swarm holds a single vector that contains the three PID gains and every uncertain parameter, as listed in eq. (15).

At every iteration, the algorithm changes the velocity and then the position of each particle. One part of the search looks for the NCS (Nonlinear Control Systems) parameter set that gives the largest tracking error. The other part looks for the trio (M_r,M_k,M_f) that gives the smallest error.

Each particle moves toward its own best previous combination of gains and parameters and also toward the swarm's best combination. The parameter set produces the worst error and the gain set that produces the least error. Equation (15) provides the fitness value for any candidate vector.

The task is a min–max problem. The outer loop minimizes the cost with respect to M_r,M_k,M_f, where the inner loop maximizes the same cost. The compact statement of this min–max outcome appears in eq. (16). Running the PID controller for every proposed parameter vector demands a large amount of CPU time. A surrogate model trained offline with neural networks replaces most of those runs.

Once trained, the surrogate gives an approximate value of $(M_r,M_k,M_f,\varnothing_c)$ almost instantly – GA or PSO can test far more candidates while searching the combined space of gains. The surrogate function $\hat{\mathrm{J}}$ therefore stands in for the exact cost during the optimization:

$$L(M_r,M_k,M_f,\varnothing_c) \approx L(M_r,M_k,M_f,\varnothing_c) \tag{16}$$

Each adversarial training run serves as a separate experiment. The agent receives a single numerical reward after each worst-case test, and this reward equals the measured performance of the PID loop during that test. The sequence of rewards drives the agent's learning in eq. (17). Adversarial machine learning methods create extreme situations on purpose. Each situation is built to stress a PID controller as much as possible. A model trained with those methods learns which settings give the poorest controller response. After training, the model offers the hardest cases while the controller is tuned, and the final PID settings withstand tougher conditions:

$$t(v) = \max_{\varnothing_c} L(M_r,M_\mathrm{k},M_f,\varnothing_c) \tag{17}$$

C. ENSEMBLE MODEL

Figure 2 illustrates the complete offline-to-online pipeline, from scenario generation and IGA/IPSO optimization through repository

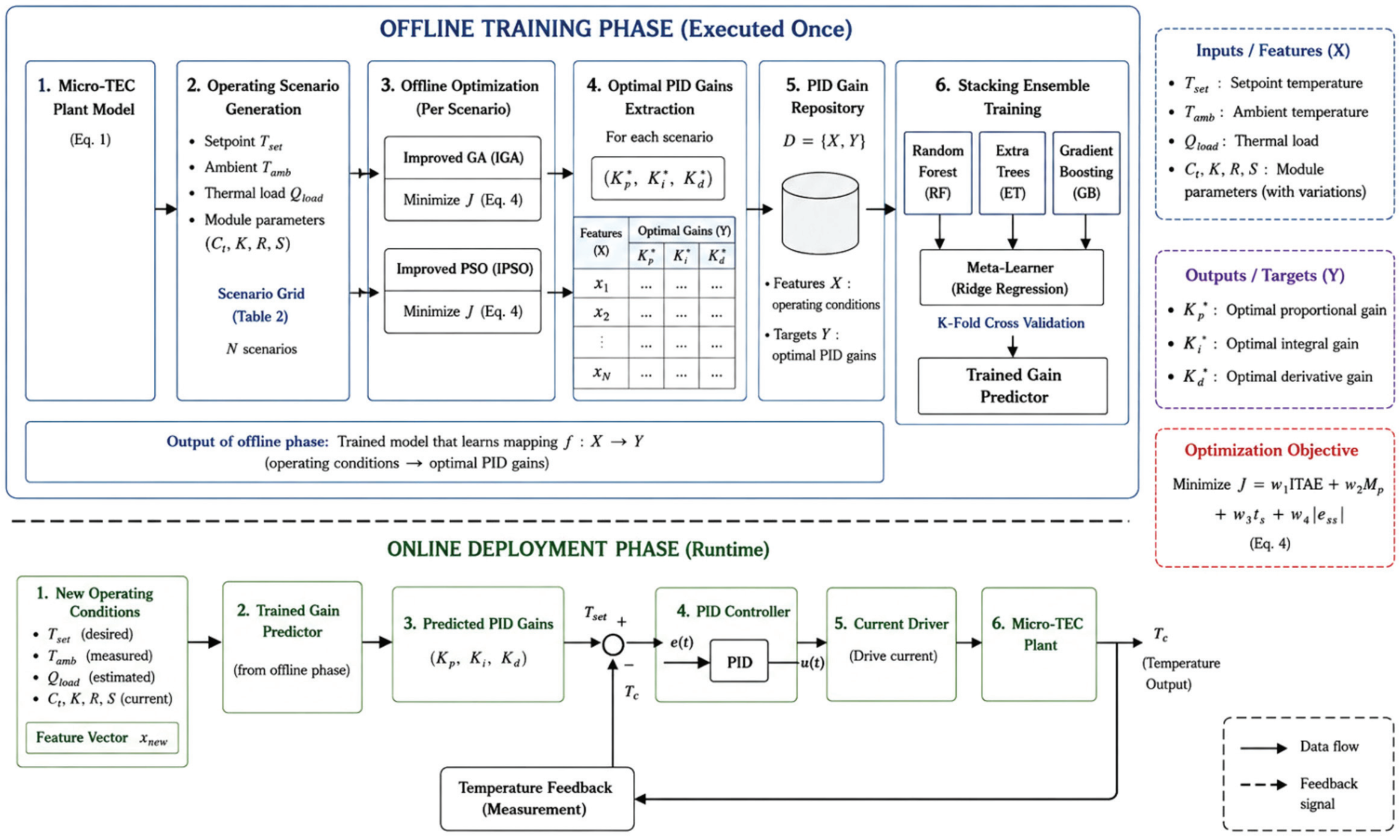

Fig. 2. Proposed ensemble.

construction and stacking-ensemble training, to online gain prediction at deployment.

The proposed ensemble learning module replaces repeated evolutionary optimization by learning the mapping between operating scenarios and near-optimal PID gains. First, improvised IGA and IPSO are executed offline across multiple scenarios to construct a PID tuning knowledge repository. In case of each scenario $T_j = [g_0,g_1,g_2,g_3]$, the optimal performance gain $C_j^* = [M_r,M_k,M_f]$ are obtained through control objective K; this gives te supervised pairs (T_j,C_j^*).

To approximate $G(.)$, ensemble learning techniques (Random Forest, Extra Trees, and Gradient Boosting) are trained, and their predictions are combined through stacking:

$$C'' = h(C''_{RF},C''_{ET},C''_{GB})$$

Thus, the ensemble distills optimization knowledge generated by IGA and IPSO into a fast predictive model, enabling real-time, scenario-aware PID tuning without online evolutionary search.

IV. PERFORMANCE EVALUATION

A. EXPERIMENTAL SETUP

All controllers are tested on the Micro-TEC plant of Section III. Temperature is given in degrees Celsius and time in seconds, so the results describe Micro-TEC thermal behavior rather than a generic step response. Five controllers are compared: GA-tuned PID, PSO-tuned PID, ML-assisted GA-PID, ML-assisted PSO-PID, and the proposed ensemble-scheduled PID. The reported metrics are averaged over 375 test scenarios that were held out from training and validation; the repository holds 2,500 scenarios, split into 1,750 for training, 375 for validation, and 375 for testing. Setpoint tracking is tested at 25, 40, 60, and 95 C. Disturbance rejection uses a thermal-load step of +1 W at 100 s and −1 W at 200 s, and robustness is checked for an ambient change from 25 C to 35 C.

Table I summarizes the resulting control-performance metrics, and Table II reports the ensemble's gain-prediction accuracy and offline efficiency.

B. RESULTS

Figure 3 illustrates the tracking performance of the optimized gain-scheduled PID controller under varying reference conditions. The target output represents the desired setpoint trajectory, while the system response shows the actual plant output obtained using the tuned PID gains. As observed, the controller is able to closely follow setpoint changes with minimal overshoot and smooth transient behavior. The response demonstrates fast convergence to the desired value after each setpoint transition, indicating effective tuning of the PID parameters. The absence of sustained oscillations and the reduced settling time highlight the stability and robustness achieved through the learning-assisted PID tuning approach. The results confirm that the optimized gain-scheduled PID controller maintains accurate tracking performance across multiple operating regions.

Fig. 4 illustrates the system response obtained using a GA-optimized PID controller in the presence of external disturbances. Despite the injected disturbances, the controller maintains stable tracking of the reference signal with limited deviation from the

Table I. Micro-TEC control performance on held-out test scenarios (mean ± std)

Controller	OS (%)	ts (s)	SSE (C)	Tune (ms)
GA-PID	6.95 ± 0.42	18.2 ± 0.9	0.31 ± 0.03	1240
PSO-PID	6.31 ± 0.39	16.8 ± 0.8	0.27 ± 0.02	980
ML-GA-PID	5.24 ± 0.35	14.6 ± 0.7	0.21 ± 0.02	88
ML-PSO-PID	4.88 ± 0.31	13.9 ± 0.6	0.18 ± 0.02	74
Ensemble (proposed)	3.92 ± 0.25	12.1 ± 0.5	0.11 ± 0.01	15

As shown in Table I, compared with the standalone PSO-tuned PID, the proposed ensemble controller lowers overshoot by 37.9%, settling time by 28.0%, and steady-state error by 59.3%, and it selects gains about 65 times faster (15 ms against 980 ms). As reported in Table II, the ensemble predicts the three gains with an R-squared of at least 0.985, and the surrogate-assisted offline search removes 72.3% of the full closed-loop simulations.

Table II. Ensemble prediction accuracy and repository/efficiency metrics

Predicted gain	*R*-squared	MAE
Kp	0.991	0.12
Ki	0.988	0.05
Kd	0.985	0.03

Repository/efficiency metric	Value
Total scenarios	2,500
Train/validation/test	1,750/375/375
Full simulations saved	72.3%
Online prediction time	15 ms

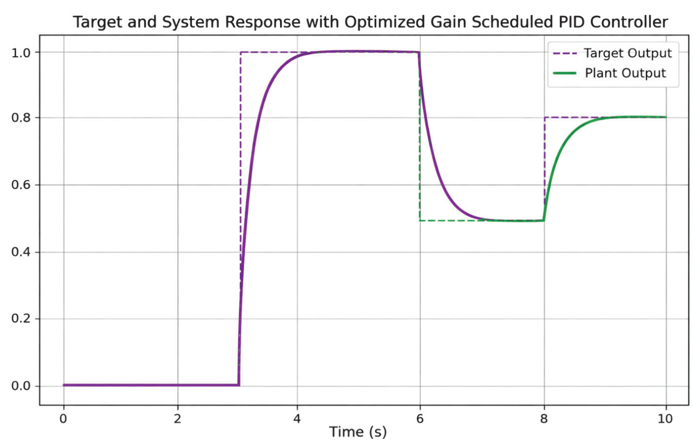

Fig. 3. Target and system response.

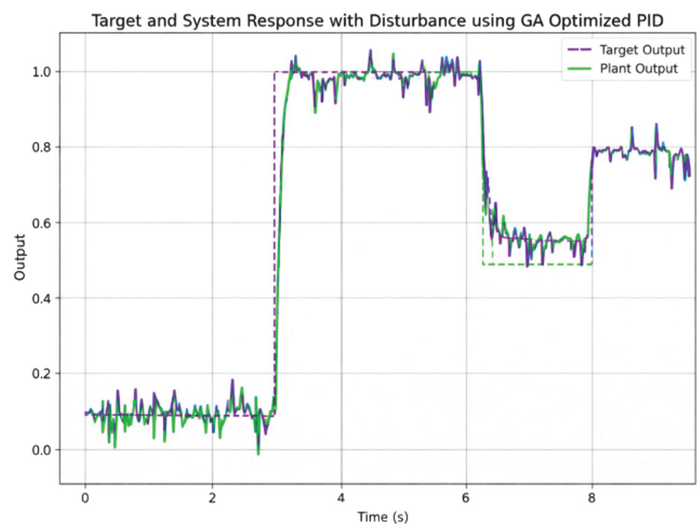

Fig. 4. GA-optimized PID controller in the presence of external disturbances.

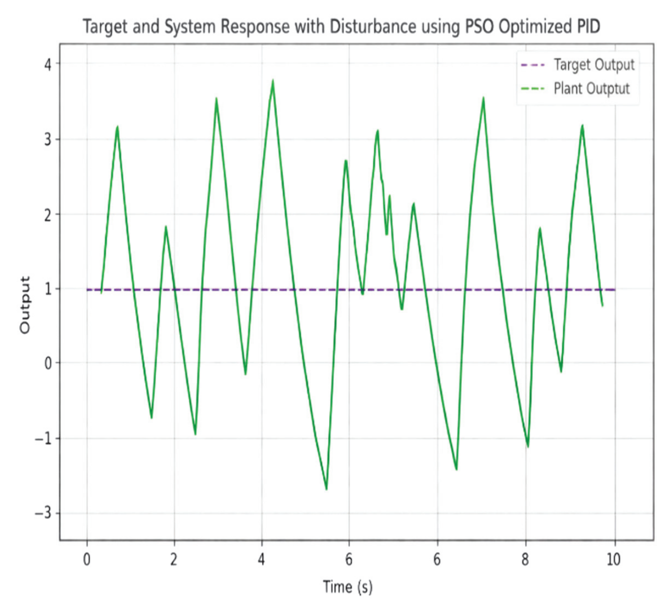

Fig. 5. PSO-optimized PID controller subjected to external disturbances.

desired output. Transient fluctuations are observed during disturbance intervals; however, the response quickly recovers and converges to the target value. This behavior demonstrates the robustness of the GA-tuned PID controller and its ability to mitigate disturbance effects while preserving overall stability.

Fig. 5 presents the response of the PSO-optimized PID controller subjected to external disturbances. Compared to the reference trajectory, the plant output exhibits larger oscillations and increased sensitivity to disturbances, indicating reduced disturbance rejection capability. Although the controller responds rapidly to setpoint changes, sustained oscillatory behavior suggests that the PSO-tuned gains are less effective in maintaining stability under adverse conditions. This result shows the importance of careful tuning strategies when applying optimization-based PID controllers in disturbance-prone environments.

Fig. 6 presents a comparative evaluation of PID tuning strategies under identical reference conditions. The GA- and PSO-optimized PID controllers exhibit noticeable oscillations

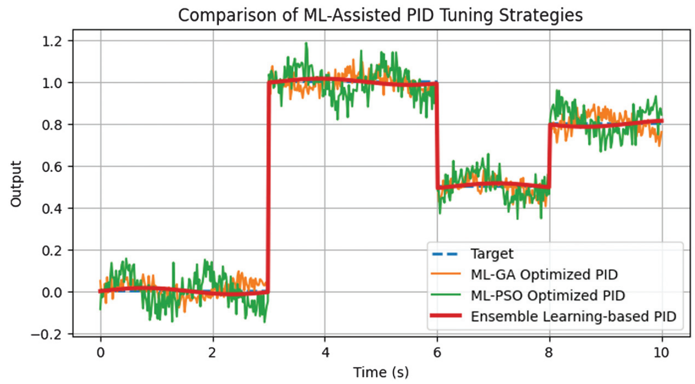

Fig. 6. PID tuning strategies under identical reference conditions.

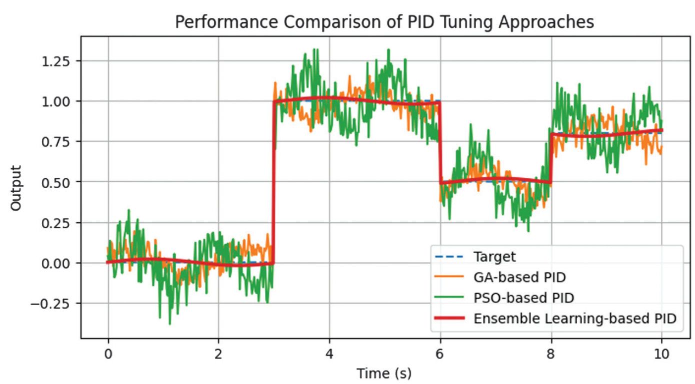

Fig. 7. GA-optimized PID, ML-assisted PSO-optimized PID, and the proposed ensemble learning-based PID controller under identical reference conditions.

and sensitivity to disturbances, particularly during setpoint transitions. In contrast, the ensemble learning-based PID controller demonstrates significantly improved tracking accuracy, reduced oscillatory behavior, and faster convergence to the desired setpoint. The smoother response and minimal steady-state deviation indicate superior robustness and generalization capability of the ensemble approach, confirming its effectiveness for data-driven PID tuning across varying operating regions.

Fig. 7 compares the tracking performance of ML-assisted GA-optimized PID, ML-assisted PSO-optimized PID, and the proposed ensemble learning-based PID controller under identical reference conditions. While both ML-GA and ML-PSO controllers improve tracking accuracy compared to classical tuning, noticeable oscillations and transient fluctuations remain during setpoint transitions. In contrast, the ensemble learning-based PID controller exhibits smoother response characteristics, minimal overshoot, and faster settling across all operating regions. The close alignment between the ensemble response and the target trajectory highlights its superior generalization capability and robustness, validating the effectiveness of the ensemble approach for data-driven PID tuning.

V. CONCLUSION

This work presented an ensemble-based PID gain learning framework for Micro-TEC temperature control applications. While evolutionary algorithms such as GA and PSO could improve PID tuning performance, their repeated execution under varying operating conditions introduces significant computational overhead. To address this limitation, the proposed approach uses offline optimization to construct a PID gain knowledge repository and subsequently employs ensemble learning to approximate the mapping between operating scenarios and optimal controller parameters. On the Micro-TEC plant, the controller brought overshoot down to 3.92% (37.9% below the PSO-tuned PID), settling time to 12.1 s (a 28.0% reduction), and steady-state error to 0.11 C (a 59.3% reduction), and it produced its gains in 15 ms, roughly 65 times faster than PSO. The method was best understood as an offline-trained gain scheduler rather than a fully online adaptive controller. The gains were produced instantly at run time, but the model itself stayed fixed after training.

By integrating multiple regression learners through a stacking strategy, the ensemble model captured nonlinear relationships between plant conditions and suitable gain values while maintaining low computational cost during real-time operation. Comparative analysis demonstrated that the ensemble-based controller provided smoother transient response, reduced oscillatory behavior, improved disturbance rejection, and enhanced generalization capability compared to individually optimized controllers.

The proposed framework preserved the simplicity and interpretability of classical PID control while augmenting it with data-driven adaptability. This combination offered a practical pathway for scalable and efficient gain scheduling in nonlinear thermal systems. Future work may explore incremental learning strategies, hardware-level validation, and integration with edge-deployed control platforms to further enhance real-world applicability.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] B. Hekimoğlu, "Optimal tuning of fractional order PID controller for DC motor speed control via chaotic Atom Search Optimization Algorithm," *IEEE Access*, vol. 7, pp. 38100–38114, 2019, doi: 10.1109/ACCESS.2019.2905961.

[2] B. Guo *et al.*, "Optimal design and simulation for PID Controller Using Fractional-Order Fish Migration Optimization Algorithm," *IEEE Access*, vol. 9, pp. 8808–8819, 2021, doi: 10.1109/ACCESS.2021.3049421.

[3] P. Zhang *et al.*, "Boosting learning efficiency in few-shot tasks with layer-adaptive PID control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 48, no. 6, pp. 7079–7096, 2026, doi: 10.1109/TPAMI.2026.3663608.

[4] J. Zeng and W. Qiao, "A linear machine learning-based model predictive control with repetitive and PI elements for a three-phase inverter," *IEEE Trans. Ind. Appl.*, vol. 61, no. 6, pp. 9529–9539, Nov-Dec. 2025, doi: 10.1109/TIA.2025.3579446

[5] W. Li *et al.*, "Human-following control method based on adaptive recurrent PID controller with self-tuning filter," *IEEE Trans. Hum.-Mach. Syst.*, vol. 55, no. 1, pp. 48–57, Feb. 2025, doi: 10.1109/THMS.2024.3515045.

[6] Y. S. Arikuşu and N. Bayhan, "Design of a novel PID controller based on machine learning algorithm for a micro-thermoelectric cooler of the polymerase chain reaction device," *IEEE Access*, vol. 12, pp. 61959–61977, 2024, doi: 10.1109/ACCESS.2024.3392734

[7] P. Grelewicz *et al.*, "Application of machine learning to performance assessment for a class of PID-based control systems," *IEEE Trans. Syst. Man Cybern: Syst.*, vol. 53, no. 7, pp. 4226–4238, July 2023, doi: 10.1109/TSMC.2023.3244714

[8] M. Zhou *et al.*, "PID controller-inspired model design for single image de-raining," *IEEE Trans. Circuits Syst. II Exp. Briefs*, vol. 69, no. 4, pp. 2351–2355, April 2022, doi: 10.1109/TCSII.2021.3137935.

[9] R. Muduli, D. Jena, and T. Moger, "Application of reinforcement learning-based adaptive PID controller for automatic generation control of multi-area power system," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 1057–1068, 2025, doi: 10.1109/TASE.2024.3359219

[10] A. A. Al Barazi and E. Aykut, "Automated tuning of PID controllers using a generalized discrete PID structure with particle swarm optimization," *IEEE Access*, vol. 13, pp. 185994–186013, 2025, doi: 10.1109/ACCESS.2025.3625055

[11] W. Fu, Y. Liu, and X. Zhang, "Research on accurate motion trajectory control method of four-wheel steering AGV based on stanley-PID control," *Sensors*, vol. 23, no. 16, p. 7219, Aug. 2023.

[12] J. Feng *et al.*, "Path planning and trajectory tracking for autonomous obstacle avoidance in automated guided vehicles at automated terminals," *Axioms*, vol. 13, no. 1, p. 27, Dec. 2023.

[13] W. P. N. D. Reis, G. E. Couto, and O. M. Junior, "Automated guided vehicles position control: A systematic literature review," *J. Intell. Manuf.*,vol. 34, no. 4, pp. 1483–1545, Apr. 2023.

[14] J. Shang, J. Zhang, and C. Li, "Trajectory tracking control of AGV based on time-varying state feedback," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, p. 162, Dec. 2021.

[15] Y. Zhang *et al.*, "Research on path planning and path tracking control of autonomous vehicles based on improved APF and SMC," *Sensors*, vol. 23, no. 18, p. 7918, Sep. 2023.

[16] M. Aizat, A. Azmin, and W. Rahiman, "A survey on navigation approaches for automated guided vehicle robots in dynamic surrounding," *IEEE Access*, vol. 11, pp. 33934–33955, 2023.

[17] A. C. Nearchou, "Adaptive navigation of autonomous vehicles using evolutionary algorithms," *Artif. Intell. Eng.*, vol. 13, no. 2, pp. 159–173, Apr. 1999.

[18] M. De Ryck, M. Versteyhe, and F. Debrouwere, "Automated guided vehicle systems, state-of-the-art control algorithms and techniques," *J. Manuf. Syst.*, vol. 54, pp. 152–173, Jan. 2020.

[19] H. Kanoh, "Dynamic route planning for car navigation systems using virus genetic algorithms," *Int. J. Knowl-based Intell. Eng. Syst.*, vol. 11, no. 1, pp. 65–78, Feb. 2007.

[20] Z. Yu, Y. Qu, and Y. Zhang, "Distributed fault-tolerant cooperative control for multi-UAVs under actuator fault and input saturation," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 6, pp. 2417–2429, Nov. 2019.

[21] Z. Yu *et al.*, "Distributed finite-time fault-tolerant containment control for multiple unmanned aerial vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2077–2091, Jun. 2020.

[22] C. Qu *et al.*, "A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning," *Knowl-Based Syst.*, vol. 194, art. no. 105530, 2020, doi:10.1016/j.knosys.2020.105530

[23] F. Han *et al.*, "Multi-objective particle swarm optimization with adaptive strategies for feature selection," *Swarm. Evol. Comput.*, vol. 62, art. no. 100847, 2021, doi: 10.1016/j.swevo.2021.100847

[24] S. K. Joshi and J. C. Bansal, "Parameter tuning for meta-heuristics," *Knowl-Based Syst.*, vol. 189, p. 105094, 2020, doi:10.1016/j.knosys.2019.105094.

[25] J. Li, Y. Yuan, and X. Luo, "Learning error refinement in stochastic gradient descent-based latent factor analysis via diversified PID controllers," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 9, no. 5, pp. 3582–3597, Oct. 2025, doi: 10.1109/TETCI.2025.3547854.