Performance comparison of SSA, GWO, and ISSA for PID tuning in leader-follower mobile robot path following

Le Thi Minh Tam[1], Nguyen Thi Thin[2], Nguyen Viet Ngu[1*]

[1]Hung Yen University of Technology and Education
[2]North Mecha Electric and Water Resources College
**Corresponding author Email: ngunguyenviet77@utehy.edu.vn*
DOI: https://doi.org/10.64032/mca.v30i3.430

Abstract

This paper compares GWO, SSA, and ISSA for PID tuning in leader–follower mobile robot circular path following under noise and disturbances. Using a unicycle model and a multi-objective cost function, all three methods outperform the manually tuned PID baseline and achieve stable circular motion after the transient stage. In representative single-run simulations, GWO-PID and SSA-PID show faster transient response and slightly better follower tracking than ISSA-PID, although ISSA remains competitive. A 20 seeds evaluation further shows that SSA and ISSA obtain lower mean objective values than GWO, with SSA giving the most stable performance. However, angular velocity oscillations remain for all methods, indicating the need for smoother control and broader validation.

***Keywords**: Circular Path Following; Formation Control; GWO; ISSA; Leader–Follower Mobile Robots; PID Tuning; SSA; Swarm Intelligence*

1. Introduction

Mobile robots have become an essential technology in modern automation due to their mobility, flexibility, and capability to operate in uncertain and dynamic environments. They are widely used in industrial logistics, warehouse automation, inspection, surveillance, and cooperative transportation, where autonomous navigation and coordinated motion are critical [1]. In particular, leader–follower multi-robot systems have attracted growing attention because they enable scalable cooperation and distributed task execution under uncertainty [2]. However, accurate path following for nonholonomic mobile robots remains a challenging problem because of nonlinear kinematics, actuator saturation, sensing noise, model mismatch, and the coupling between translational and rotational motion [3].

Among typical reference paths, circular trajectories are especially important because they represent many practical maneuvers such as turning, orbiting, perimeter patrol, and coordinated inspection. Compared with straight-line motion, circular path tracking is more difficult since the robot must continuously regulate both linear and angular velocities along a curved path. In leader–follower systems, this challenge is further amplified because the follower must track the leader while maintaining the desired formation distance. Prior studies have shown that curvature-dependent effects can significantly degrade tracking accuracy and motion smoothness, especially at higher speeds or under disturbances [4].

To solve path following problems, classical control methods such as PID and model predictive control (MPC) are widely used. PID remains attractive because of its simple structure, ease of implementation, and low computational cost, and it can provide effective disturbance rejection when properly tuned [5]. However, PID performance is highly sensitive to gain selection, and fixed gains may lead to overshoot, oscillation, or reduced robustness under varying operating conditions. By contrast, MPC offers a systematic framework for handling multivariable dynamics and actuator constraints, and has demonstrated strong path following capability [6]. It has also been applied to leader–follower formations with good stability properties [7]. Nevertheless, MPC typically requires a more accurate model and higher computational cost, which may limit its use in embedded robotic platforms.

As a result, metaheuristic optimization has emerged as a promising alternative for controller tuning. Grey Wolf Optimizer (GWO) is widely recognized for its simple structure and good balance between exploration and exploitation [8], while improved GWO strategies have shown stronger convergence and robustness in engineering optimization problems [9]. Similarly, Sparrow Search Algorithm (SSA) has demonstrated competitive search capability and strong optimization performance in both theoretical and benchmark studies [10], and its recent variants have received increasing attention for complex optimization tasks [11]. To further improve performance, Improved Sparrow Search Algorithm (ISSA) schemes incorporating chaotic initialization, opposition-based learning, and adaptive search strategies have been proposed to enhance global exploration and avoid local optima [12]. Such adaptive mechanisms are known to improve the balance between exploration and exploitation in metaheuristic algorithms [13]. These developments have encouraged the use of ISSA in robotic applications such as mobile robot path planning in complex environments [14]. In parallel, optimization-based PID tuning has shown clear benefits in robotic systems by improving transient response and steady-state accuracy compared with manual tuning [15].

Despite these advances, an important research gap remains. Although extensive research exists on generic controller tuning and global path planning, there remains a notable lack of comprehensive studies directly comparing the efficacy of SSA, GWO, and ISSA for PID tuning specifically in the context of leader-follower mobile robot path following. In addition, many studies emphasize endpoint accuracy or

Received:16 April 2026; Revised: 01 June 2026; Accepted: 24 June 2026.

optimizer convergence, but pay less attention to motion smoothness, oscillatory behavior, and cumulative tracking error on curved trajectories. For disturbed multi-robot systems, multi-objective formulations that jointly consider tracking accuracy, formation preservation, and control effort are increasingly necessary [16]. Recent research has also highlighted the need to bridge the gap between global optimization performance and local tracking quality [17], since practical path following should be evaluated not only by accuracy but also by smoothness and stability [18].

Motivated by these issues, this paper presents a comparative study of GWO, SSA, and ISSA for PID tuning in leader–follower mobile robot circular path following under noise and disturbances. A disturbance-aware multi-objective cost function is used to optimize the PID gains by considering tracking accuracy, formation maintenance, heading mismatch, and control effort. The main contributions are: (i) a unified benchmark for comparing GWO, SSA, and ISSA under identical settings; (ii) a multi-objective PID tuning framework for leader–follower circular tracking; and (iii) a combined evaluation using representative simulations and 20-run statistical analysis. The remainder of this paper is organized as follows. Section 2 presents the Problem Formulation. Section 3 describes the Methodology. Section 4 introduces the Numerical simulation setup. Section 5 discusses the Results and Discussion. Finally, Section 6 concludes the paper.

2. Problem formulation

2.1 System description and kinematic model

Consider a leader–follower mobile robot system composed of two nonholonomic unicycle robots, indexed by $i \in \{L, F\}$ where L and F denote the leader and follower, respectively. The state and control input of robot (i) are defined as follows:

$$\mathbf{x}_i(t) = [x_i(t), y_i(t), \theta_i(t)]^T \tag{1}$$

$$\mathbf{u}_i(t) = [v_i(t), \omega_i(t)]^T \tag{2}$$

where x_i and y_i denote the planar coordinates, θ_i is the heading angle, and v_i and ω_i are the linear and angular velocities, respectively.

The continuous-time kinematic model of each robot is given by:

$$\begin{aligned} \dot{x}_i(t) &= v_i(t)\cos\theta_i(t), \\ \dot{y}_i(t) &= v_i(t)\sin\theta_i(t), \\ \dot{\theta}_i(t) &= \omega_i(t). \end{aligned} \tag{3}$$

In the numerical implementation, the system is propagated using forward Euler discretization with sampling time T_s. The corresponding discrete-time update equations are:

$$x_i(k+1) = x_i(k) + T_s v_i^a(k)\cos\theta_i(k) \tag{4}$$

$$y_i(k+1) = y_i(k) + T_s v_i^a(k)\sin\theta_i(k) \tag{5}$$

$$\theta_i(k+1) = wrap(\theta_i(k) + T_s\omega_i^a(k)) \tag{6}$$

where $\mathrm{wrap}(\cdot)$ maps an angle to the interval $(-\pi, \pi]$

To reflect actuator uncertainty and physical input limits, the executed linear and angular velocities are modeled as disturbed and saturated versions of the commanded inputs:

$$v_i^a(k) = clip(v_i(k) + \eta_{v,i}(k), -v_{max}, v_{max}) \tag{7}$$

$$\omega_i^a(k) = clip(\omega_i(k) + \eta_{\omega,i}(k), -\omega_{max}, \omega_{max}) \tag{8}$$

where $\eta_{v,i}(k)$ and $\eta_{\omega,i}(k)$ denote the disturbance terms, while $v_{\max}$ and $\omega_{\max}$ are the actuator bounds.

2.2 Circular reference generation for the leader

The leader robot is required to follow a circular path of radius R centered at (x_c, y_c). Rather than using a purely time-parameterized reference, the controller computes a virtual target point on the desired circle from the measured leader position and a look-ahead mechanism.

Let the measured leader state at time step k be:

$$\mathbf{x}_L(k) = [\tilde{x}_L(k), \tilde{y}_L(k), \tilde{\theta}_L(k)]^T \tag{9}$$

The angular location of the leader on the circle is estimated from the noisy measured position, and a look-ahead angle δ_{la} is added to improve tracking behavior:

$$\phi_L(k) = atan2(\tilde{y}_L(k) - y_c, \tilde{x}_L(k) - x_c) + \delta_{la} \tag{10}$$

The resulting leader reference state is then defined as:

$$x_{L,ref}(k) = x_c + R\cos\phi_L(k) \tag{11}$$

$$y_{L,ref}(k) = y_c + R\sin\phi_L(k) \tag{12}$$

$$\theta_{L,ref}(k) = wrap(\phi_L(k) + \pi/2) \tag{13}$$

In addition, the circular reference includes feedforward linear and angular velocity terms corresponding to the prescribed circular motion:

$$v_{L,ff} = R\omega_c, \quad \omega_{L,ff} = \omega_c \tag{14}$$

This construction ensures that the leader reference remains geometrically consistent with the target circle and that the reference heading is tangent to the path.

2.3 Reference eneration for the follower

The follower robot is not assigned an independent global path. Instead, it is required to maintain a desired spacing behind the leader while following the leader motion.

Accordingly, the follower reference is generated directly from the leader state. In the implementation, the leader position used in this step is affected by measurement noise, while the leader heading is taken from the current leader state. The follower reference is defined as:

$$x_{F,ref}(k) = \bar{x}_L(k) - d_{des}\cos\bar{\theta}_L(k) \tag{15}$$

$$y_{F,ref}(k) = \bar{y}_L(k) - d_{des}\sin\bar{\theta}_L(k) \tag{16}$$

$$\theta_{F,ref}(k) = \bar{\theta}_L(k) \tag{17}$$

To improve consistency with the leader motion, the follower reference also includes feedforward velocity terms obtained from the leader control signal:

$$\mathrm{v}_{\mathrm{F,ff}}(\mathrm{k}) = \mathrm{clip}(\mathrm{v}_{\mathrm{L}}(\mathrm{k}), 0, \mathrm{v}_{\max}) \tag{18}$$

$$\omega_{F,ff}(k) = \omega_L(k) \tag{19}$$

This rmulation converts the formation-control problem into a tracking problem with a moving reference generated online from the leader behavior.

2.4 Measurement model and performance variables

To represent sensing uncertainty, the controller uses noisy state measurements rather than exact robot states. For each robot $i \in \{L, F\}$, the measurement model is given by:

$$\tilde{x}_i(k) = x_i(k) + n_{x,i}(k), \tilde{y}_i(k) = y_i(k) + n_{y,i}(k) \quad (20)$$

$$\tilde{\theta}_i(k) = \theta_i(k) + n_{\theta,i}(k) \quad (21)$$

where $n_{x,i}(k), n_{y,i}(k)$, and $n_{\theta,i}(k)$ denote the measurement noise terms.

To evaluate the closed-loop behavior, four performance variables are considered. The leader and follower path-tracking errors are defined as the radial deviations from the target circle:

$$e_{L,path}(k) = |\sqrt{(x_L(k) - x_c)^2 + (y_L(k) - y_c)^2} - R| \quad (22)$$

$$e_{p,F}(k) = \sqrt{(x_F(k) - x_{r,F}(k))^2 + (y_F(k) - y_{r,F}(k))^2} \quad (23)$$

The formation-distance error is defined as:

$$e_d(k) = \sqrt{(x_L(k) - x_F(k))^2 + (y_L(k) - y_F(k))^2} - d_{des} \quad (24)$$

and the inter-robot heading mismatch is defined as:

$$e_\theta(k) = wrap(\theta_L(k) - \theta_F(k)) \quad (25)$$

While the leader path-following error $e_{p,L}$ evaluates the radial deviation from the desired geometric circle, the follower tracking error $e_{p,F}$ is defined as the Euclidean distance between the follower's actual position and its generated reference point ($x_{r,F}$, $y_{r,F}$). This modification ensures mathematical consistency, as the follower is commanded to maintain a specific distance behind the leader rather than strictly staying on the leader's exact circular arc.

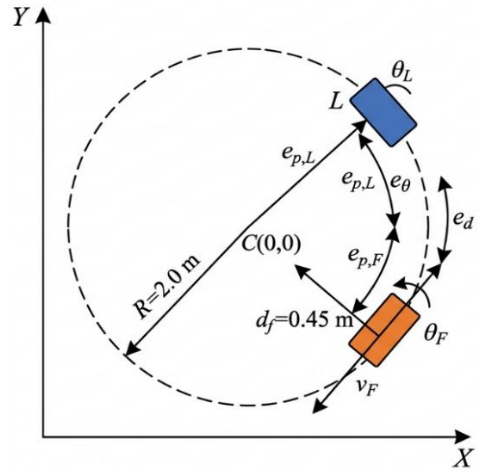

Figure 1: Leader–follower circular path following model

Table 1 is taken directly from the simulation and controller configuration used in this study.

Table 1: Robot and Simulation Parameters Used in the Problem Formulation.

Parameter	Symbol	Value
Sampling time	Δt	0.02 s
Simulation duration	T_{end}	20 s
Maximum linear velocity	V_{max}	1.2 m/s
Maximum angular velocity	Ω_{max}	(3.8) rad/s
Desired leader–follower distance	d_f	0.45 m
Circular-path radius	R	2.0 m
Circular-path angular rate	ω_r	0.20 rad/s
Circle center		(0,0) m
Position-noise standard deviation	σ_p	0.01 m
Heading-noise standard deviation	σ_θ	0.01 rad
Linear-velocity disturbance std.	σ_{d_v}	0.03 m/s
Angular-velocity disturbance std.	σ_{d_ω}	0.05 rad/s
Leader look-ahead angle	λ	0.18 rad

3. Methodology

3.1 PID tracking law

Each robot is controlled by a six-parameter PID tracking law acting on the distance-to-target and heading-to-target variables. For robot $i \in \{L, F\}$, let the measured state be:

$$\mathbf{x}_i(k) = [\tilde{x}_i(k), \tilde{y}_i(k), \tilde{\theta}_i(k)]^T \quad (26)$$

and let the reference state be:

$$\mathbf{x}_{i,ref}(k) = = [x_{i,ref}(k), y_{i,ref}(k), \theta_{i,ref}(k), v_{i,ff}(k), \omega_{i,ff}(k)]^T \quad (27)$$

where $v_{i,ff}(k)$ and $\omega_{i,ff}(k)$ denote the feedforward linear and angular velocities associated with the reference motion.

The Cartesian tracking errors are defined as:

$$e_{x,i}(k) = x_{i,ref}(k) - \tilde{x}_i(k), \quad e_{y,i}(k) = y_{i,ref}(k) - \tilde{y}_i(k) \quad (28)$$

From these quantities, the polar tracking variables are computed as:

$$\rho_i(k) = \sqrt{e_{x,i}^2(k) + e_{y,i}^2(k)} \quad (29)$$

$$\alpha_i(k) = wrap(atan2(e_{y,i}(k), e_{x,i}(k)) - \tilde{\theta}_i(k)) \quad (30)$$

Here, $\rho_i(k)$ represents the Euclidean distance to the reference point, while $\alpha_i(k)$ denotes the angular misalignment between the robot heading and the line-of-sight direction toward the reference.

The integral states are updated using anti-windup clipping:

$$I_{\rho,i}(k+1) = clip(I_{\rho,i}(k) + T_s\rho_i(k), -I_{max}, I_{max}) \quad (31)$$

$$I_{\alpha,i}(k+1) = clip(I_{\alpha,i}(k) + T_s\alpha_i(k), -I_{max}, I_{max}) \quad (32)$$

and the derivative terms are approximated numerically by:

$$D_{\rho,i}(k) = (\rho_i(k) - \rho_i(k-1)) / T_s, \quad D_{\alpha,i}(k) = (\alpha_i(k) - \alpha_i(k-1)) / T_s \quad (33)$$

The commanded linear and angular velocities are then computed as:

$$v_i(k) = (v_{i,ff}(k) + K_{p,\rho}\rho_i(k) + \\ + K_{i,\rho}I_{\rho,i}(k) + K_{d,\rho}D_{\rho,i}(k))\cos\alpha_i(k) \quad (34)$$

$$\omega_i(k) = \omega_{i,ff}(k) + K_{p,\alpha}\alpha_i(k) + \\ + K_{i,\alpha}I_{\alpha,i}(k) + K_{d,\alpha}D_{\alpha,i}(k) \quad (35)$$

The multiplicative factor $\cos\alpha_i(k)$ in the linear-velocity channel reduces aggressive forward motion when the robot is strongly misaligned with the target direction. The gain vector to be optimized is:

$$\mathbf{K} = [K_{p,\rho}, K_{i,\rho}, K_{d,\rho}, K_{p,\alpha}, K_{i,\alpha}, K_{d,\alpha}]^T \quad (36)$$

For consistency, and crucially to limit the search dimension, a shared gain vector is assigned to both the leader and follower controllers. While the leader and follower perform slightly different tasks, decoupling their controllers would expand the search space from 6 to 12 dimensions. Given the limited population size (10) and iteration budget (15) chosen to minimize computational overhead, a higher-dimensional space would significantly degrade the convergence reliability of the metaheuristic algorithms.

3.2 Optimization problem for PID gain tuning

The PID tuning problem is formulated as a bounded nonlinear optimization problem: $\min_{\mathbf{K}} J(\mathbf{K})$ (37)

subject to: $\mathbf{K}_{min} \le \mathbf{K} \le \mathbf{K}_{max}$ (38)

where $\mathbf{K}_{\min}$ *and* $\mathbf{K}_{\max}$ denote the lower and upper bounds of the feasible search space.

To reduce sensitivity to a single initial condition, the objective is evaluated over three simulation scenarios with different initial leader–follower poses. Let $J_s(\mathbf{K})$ denote the scenario-dependent cost. The overall objective is defined as the arithmetic mean:

$$J(\mathbf{K}) = (1/3)\sum_{s=1}^{3} J_s(\mathbf{K}) \quad (39)$$

For each scenario, the cost function combines path-tracking accuracy, formation preservation, heading consistency, control effort, and worst-case radial deviation from the target circle:

$$J_s = w_1\bar{J}_{path,L} + w_2\bar{J}_{path,F} + w_3\bar{J}_{dist} + w_4\bar{J}_{\theta} + \\ w_5\bar{J}_u + w_6\max(\bar{e}_{p,L}) + w_7\max(\bar{e}_{p,F}) \quad (40)$$

To eliminate dimensional bias caused by mixing different physical units (meters, radians), all terms in the multi-objective function are normalized using their respective maximum allowable thresholds before applying the weights. The weighting coefficients are selected based on task priority: the highest weights are assigned to formation distance maintenance w_3 and leader path accuracy w_1, w_6 to ensure the primary mission goals are strictly met, while lower weights are assigned to heading mismatch w_4 and control effort w_6 to serve as secondary regularization terms.

The control-effort penalty is defined as:

$$J_u = 0.01T_s\sum_{k=1}^{N}(v_L^2(k) + \\ +0.3\omega_L^2(k) + v_F^2(k) + 0.3\omega_F^2(k)) \quad (41)$$

It must be noted that this control-effort penalty is a heuristic mathematical proxy designed solely to limit aggressive velocity commands and prevent actuator saturation. It does not represent actual physical energy consumption, as the system model does not incorporate detailed actuator dynamics or power consumption profiles.

Additionally, the root-mean-square operator is defined as:

$$RMS(e) = \sqrt{(1/N)\sum_{k=1}^{N} e^2(k)} \quad (42)$$

This objective structure gives greater importance to formation maintenance and path-tracking accuracy while still penalizing excessive actuation and large transient deviations.

3.3 Grey wolf optimizer

GWO is employed as the first benchmark algorithm for solving the PID tuning problem. Let $\mathbf{X}_i^t$ denote the i-th candidate solution at iteration t, where each candidate represents a PID gain vector.

After evaluating the objective value of the population, the three best candidate solutions are denoted by $\mathbf{X}_\alpha$, $\mathbf{X}_\beta$ and $\mathbf{X}_\delta$. For each wolf, three intermediate position vectors are computed as:

$$\mathbf{X}_1 = \mathbf{X}_\alpha - \mathbf{A}_1 \odot |\mathbf{C}_1 \odot \mathbf{X}_\alpha - \mathbf{X}_i^t| \quad (43)$$

$$\mathbf{X}_2 = \mathbf{X}_\beta - \mathbf{A}_2 \odot |\mathbf{C}_2 \odot \mathbf{X}_\beta - \mathbf{X}_i^t| \quad (44)$$

$$\mathbf{X}_3 = \mathbf{X}_\delta - \mathbf{A}_3 \odot |\mathbf{C}_3 \odot \mathbf{X}_\delta - \mathbf{X}_i^t| \quad (45)$$

where $\odot$ denotes elementwise multiplication, and the coefficient vectors are defined by:

$$\mathbf{A}_m = 2a\mathbf{r}_{1,m} - a, \mathbf{C}_m = 2\mathbf{r}_{2,m}, m \in \{1,2,3\} \quad (46)$$

The exploration parameter decreases linearly with the iteration index according to: $a = 2 - 2t/T_{\max}$ (47)

The updated wolf position is then obtained by averaging the three intermediate vectors and projecting the result onto the feasible search interval:

$$\mathbf{X}_i^{t+1} = clip((\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3)/3, \mathbf{K}_{min}, \mathbf{K}_{max}) \quad (48)$$

In this study, GWO is executed with the same population size and iteration budget used for the other benchmark methods.

3.4 Sparrow search algorithm

SSA is used as the second benchmark optimizer. At each iteration, the population is divided into producers, scroungers, and warners. Let PD and SD denote the numbers of producers and warners, respectively.

For the producer group, the position update is given by:

$$\mathbf{X}_i^{t+1} = \begin{cases} \mathbf{X}_i^t \odot \exp\left(-\dfrac{i}{\alpha_t T_{\max} + \varepsilon}\right), R_2 < ST \\ \mathbf{X}_i^t + \mathbf{Q}_i, \quad R_2 \ge ST \end{cases} \quad i = 1,\ldots,n_P \quad (49)$$

where R_2 is the warning indicator, ST is the safety threshold, $\mathbf{Q}_i$ is a Gaussian random vector, ε is a small positive constant, and α_t is the producer scaling coefficient. In standard SSA, $\alpha_t = 1$.

For the scrounger group, the position update is defined as:

$$\mathbf{X}_i^{t+1} = \begin{cases} \mathbf{Q}_i \cdot \exp\left(\dfrac{\mathbf{X}_{\text{worst}}^t - \mathbf{X}_i^t}{i^2}\right), & i > \dfrac{n}{2} \\ \mathbf{X}_{\text{best}}^t + \left|\mathbf{X}_i^t - \mathbf{X}_{\text{best}}^t\right| \cdot \mathbf{A}_i, & i \le \dfrac{n}{2} \end{cases} \quad (50)$$

where $\mathbf{X}_{\text{best}}^t$ and $\mathbf{X}_{\text{worst}}^t$ denote the best and worst solutions at iteration t, respectively, and $\mathbf{A}_i$ is a random sign vector.

For the warner subset, the update rule is:

$$\mathbf{X}_j^{i+1} = \begin{cases} \mathbf{X}_{\text{best}}^t + \mathbf{Q}_i \cdot \left|\mathbf{X}_j^t - \mathbf{X}_{\text{best}}^t\right|, F_j^t > F_{\text{best}}^t \\ \mathbf{X}_j^t + \mathbf{U}_i \cdot \dfrac{\left|\mathbf{X}_j^t - \mathbf{X}_{\text{worst}}^t\right|}{F_j^t - F_{\text{worst}}^t + \varepsilon}, F_j^t \le F_{\text{best}}^t \end{cases} \quad (51)$$

where W denotes the index set of warners and $\mathbf{U}_i$ is a uniformly distributed random vector.

After the updates, all candidate solutions are clipped to the feasible bounds. The standard SSA implementation in this study uses uniform random initialization and does not include opposition-based learning or adaptive producer scaling.

3.5 Improved sparrow search algorithm

To enhance exploration and reduce premature convergence, an improved SSA variant is also considered. Compared with standard SSA, the ISSA implementation used in this work incorporates three additional mechanisms.

First, chaotic initialization is employed instead of direct uniform initialization. The initial chaotic variables are generated as: $z_{k+1} = \mu z_k (1 - z_k), 0 < z_k < 1$ (52)

and evolved using the logistic map:

$$X_j = K_{\min,j} + z_j \left(K_{\max,j} - K_{\min,j}\right) \quad (53)$$

The resulting chaotic sequence is then mapped to the feasible gain interval:

$$\mathbf{X}_i^{\text{opp}} = \mathbf{K}_{\min} + \mathbf{K}_{\max} - \mathbf{X}_i \quad (54)$$

This mechanism improves the diversity of the initial population.

Second, ISSA uses an adaptive producer coefficient in Eq. (49). In particular, the producer scaling term α_t decreases linearly with the iteration index, which gradually reduces exploration and strengthens exploitation as the search proceeds.

Third, opposition-based learning is applied after the standard SSA update. For each candidate solution, an opposite candidate is generated and retained if it yields a lower objective value:

$$\mathbf{X}_i^{t+1} = \begin{cases} \mathbf{X}_i^{\text{opp}}, & \text{if } J(\mathbf{X}_i^{\text{opp}}) < J(\mathbf{X}_i), \\ \mathbf{X}_i, & \text{otherwise.} \end{cases} \quad (55)$$

Consequently, ISSA combines chaotic initialization, adaptive producer control, and opposition-based learning within the same bounded optimization framework. For fairness, ISSA uses the same population size and iteration budget as GWO and SSA.

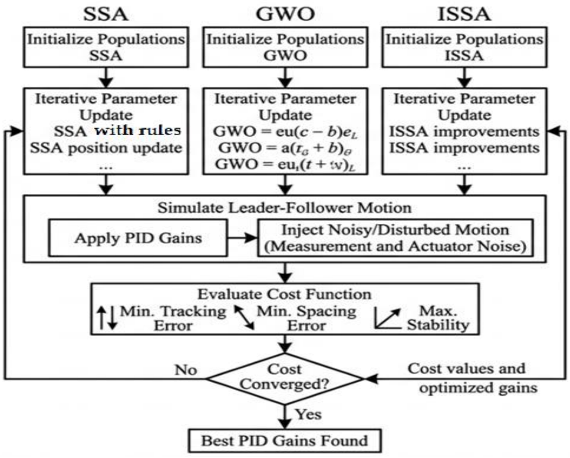

Figure 2: PID tuning and optimization framework

4. Numerical simulation setup

4.1 Simulation environment and robot settings

The simulations were conducted with a sampling time of 0.02s over 20s. Both robots were modeled as unicycle-type systems with bounded inputs, with maximum linear and angular velocities of 1.2m/s and 3.8rad/s, respectively. The leader tracked a circular reference path of radius 2.0m centered at (0, 0) with angular rate 0.20 rad/s, while the follower maintained a desired spacing of 0.45m. A look-ahead angle of 0.18rad was used to generate the leader reference point. Measurement noise and input disturbances were included to represent non-ideal operating conditions. Zero-mean Gaussian noise was added to the measured position and heading, with standard deviations of 0.01m and 0.01rad, respectively, while Gaussian disturbances with standard deviations of 0.03m/s and 0.05rad/s were applied to the executed linear and angular velocities.

4.2 Controller configuration and optimization settings

Both the leader and follower are controlled using the same six-parameter PID gain vector,

$$\mathbf{g} = \left[K_{p,\rho} \quad K_{i,\rho} \quad K_{d,\rho} \quad K_{p,\alpha} \quad K_{i,\alpha} \quad K_{d,\alpha}\right]^{\cdot}$$

which acts on the distance error ρ and heading error α defined in Section 3. The feasible search interval for the gain vector is chosen as:

$$\mathbf{g}_{\min} = [0.10 \quad 0 \quad 0 \quad 0.50 \quad 0 \quad 0]^{\cdot},$$

$$\mathbf{g}_{\max} = [5.00 \quad 1.50 \quad 1.50 \quad 12.0 \quad 2.00 \quad 2.50]^{\cdot}.$$

These bounds are selected to provide sufficient flexibility for both translational and rotational control while avoiding unrealistically large PID coefficients. Anti-windup clipping is applied to the integral states with a saturation limit of ±2.5.

To ensure a fair and rigorous statistical comparison, the exact same set of 20 random seeds was used for GWO, SSA, and ISSA during the 20-run evaluation phase. However, for the purpose of generating the single-run visual trajectory and response plots presented in Section 5, specific representative seeds (Seed 10 for GWO, 30 for SSA, and 20 for ISSA) were selected to best illustrate the typical transient characteristics of each respective optimizer. For comparison with optimized designs, a conventional manually chosen PID vector is also included: $\mathbf{g}_{\text{trad}} = [1.1 \quad 0 \quad 0.1 \quad 3.0 \quad 0 \quad 0.15]^{\top}$

This baseline is not obtained through optimization and serves only as a practical reference for assessing the benefit of the metaheuristic tuning process.

Table 2: Main Optimization and Evaluation Settings

Item	Value
Optimizers	GWO, SSA, ISSA
Population size	10
Maximum iterations	15
Conventional PID gains	[1.1, 0, 0.1, 3.0, 0, 0.15]
PID lower bounds	[0.10, 0, 0, 0.50, 0, 0]
PID upper bounds	[5.00, 1.50, 1.50, 12.0, 2.00, 2.50]
Representative seed for visualization GWO	10
Representative seed for visualization SSA	30
Representative seed for visualization ISSA	20
Objective evaluation scenarios	3
Final comparison scenario	1 leader–follower case

4.3 Training scenarios and final comparison scenario

To reduce sensitivity to a single initial condition, the optimization objective is evaluated over three different leader–follower initial configurations. The three scenario pairs used during tuning are:

$([2.0, 0.0, \pi/2], [1.5, -0.2, \pi/2])$,

$([0.0, 2.0, \pi], [-0.3, 1.5, \pi])$,

$([-2.0, 0.0, -\pi/2], [-2.4, 0.3, -\pi/2])$.

For each candidate gain vector, the closed-loop simulation is run on all three scenarios, and the final objective value is computed as the average cost over these cases. This design makes the tuning procedure less dependent on a single starting pose and encourages gains with more consistent performance across different initial geometric relationships relative to the circular path. After optimization, the resulting PID gains from GWO, SSA, and ISSA are further tested in a common comparison scenario with initial leader and follower states given by:

$\mathbf{q}_L(0) = [0, 0, \pi/2]^{\top}$, $\mathbf{q}_F(0) = [0, 1, \pi/2]^{\top}$

This final scenario is used to generate the comparative trajectory, position, velocity, path-error, formation-error, and convergence plots reported in Section 5. By using the same initial condition for all controllers in the final visualization stage, the observed differences can be attributed directly to the tuned gains rather than to differing initial configurations.

4.4 Performance metrics and output visualization

The evaluation is based on the multi-objective cost function defined in Section 3, which includes RMS leader and follower path-tracking errors, RMS formation-distance error, RMS heading mismatch, control effort, and the maximum absolute path errors of both robots. The corresponding weights emphasize formation maintenance and circular tracking while still penalizing excessive control action and large transient deviations. In addition to the objective value, the comparison also considers planar trajectories, position responses, velocity profiles, path-tracking errors, convergence curves, formation-distance errors, and multi-snapshot trajectory plots. To improve reliability, each optimizer was further evaluated over 20 independent runs. For each run, the best objective value and the associated tracking metrics were recorded, and the final results are reported as mean and standard deviation to assess robustness and repeatability.

5. Results and discussion

5.1 Results

This section first presents a statistical comparison of GWO, SSA, and ISSA over 20 independent random seeds, and then provides representative single-run trajectory and response plots for qualitative illustration. Since the three optimizers are stochastic, the 20-seed analysis is used as the primary basis for performance comparison, while the single-run figures are mainly used to visualize the closed-loop motion characteristics.

Table 3: Statistical comparison of GWO, SSA, and ISSA over 20 independent runs

Algorithm	Objective	RMS leader error (m)	RMS follower error (m)	RMS formation error (m)	RMS heading mismatch (rad)	Control effort	Max leader error (m)	Max follower error (m)
Traditional PID	4.509±0.025	0.007±0.000	0.105±0.001	0.064±0.001	0.358±0.002	0.697±0.006	0.011±0.001	0.610±0.005
GWO	4.459±0.020	0.024±0.000	0.103±0.001	0.050±0.001	0.401±0.002	1.027±0.005	0.031±0.001	0.614±0.005
SSA	4.271±0.022	0.007±0.000	0.112±0.001	0.071±0.001	0.373±0.001	0.300±0.001	0.011±0.001	0.616±0.005
ISSA	4.268±0.018	0.017±0.000	0.096±0.001	0.050±0.001	0.377±0.002	0.605±0.003	0.023±0.001	0.618±0.005

As shown in Table 3, the quantitative results demonstrate the advantage of optimization-based PID tuning over the Traditional PID baseline. The Traditional PID controller produces the highest objective value together with larger RMS tracking errors, formation errors, heading mismatch, control effort, and maximum transient deviations. Although the Traditional PID employs fixed gains, it exhibits non-zero standard deviations due to the varying random realizations of measurement noise and actuator disturbances across the 20 independent evaluation runs. In contrast, all optimized controllers (GWO, SSA, and ISSA) achieve lower mean objective values and improved tracking performance, indicating better robustness under noise and disturbance conditionsTo improve the visual interpretation of Table 3, Figures 3 and 4 present bar-chart summaries of the 20-run statistics.

Figure 3 shows the mean objective values with standard deviations, while Figure 4 compares several key metrics, including leader tracking error, follower tracking error, formation error, and control effort.

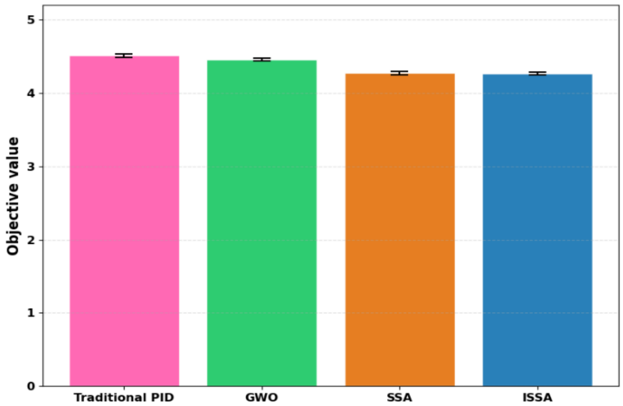

Figure 3: Mean objective with standard deviation

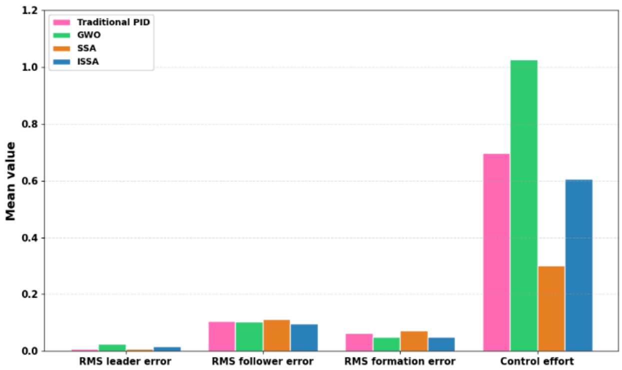

Figure 4: Grouped bar chart of key metrics

To evaluate the transient behavior and control quality in detail, a representative single-run simulation is analyzed. Table 4 presents the specific PID gain vectors obtained from the representative random seeds (Seed 10 for GWO, 30 for SSA, and 20 for ISSA) used to generate the subsequent visual responses. The conventional manually tuned gains are also included to establish a direct baseline for the trajectory and velocity profiles.

Table 4: Optimized PID Gain Vectors Obtained by GWO, SSA, and ISSA

Algorithm	$K_{p,\rho}$	$K_{i,\rho}$	$K_{d,\rho}$	$K_{p,\alpha}$	$K_{i,\alpha}$	$K_{d,\alpha}$
Traditional PID	1.1	0	0.1	3	0	0.15
GWO-PID	0.4263	0.1958	0.1551	7.0635	0.8278	0.1659
SSA-PID	0.9587	0	0	2.5296	0	0
ISSA-PID	0.8593	0.069	0	6.0136	0	0

Figure 5 shows the planar trajectories of the leader and follower robots relative to the desired circular path. All optimized PID controllers achieve stable circular motion after the transient phase and generally produce trajectories closer to the reference than the traditional PID controller. During the transient stage, GWO-PID and SSA-PID converge faster toward the desired path, whereas ISSA-PID exhibits a longer transient before reaching steady tracking.

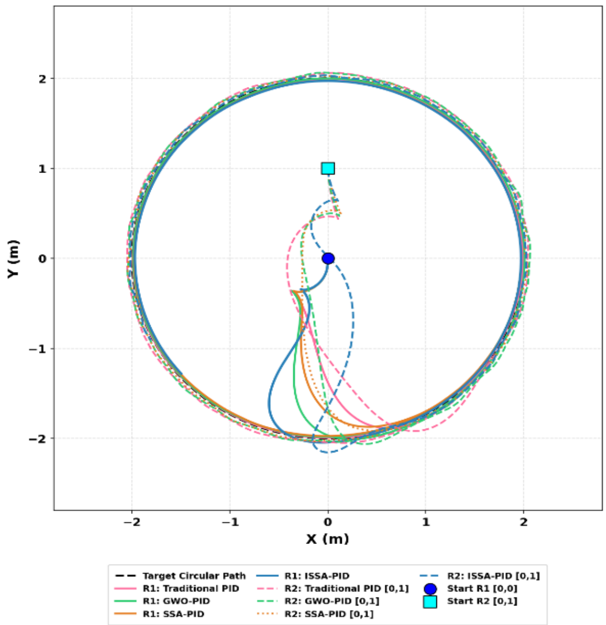

Figure 5: Trajectory comparison of leader–follower mobile robots under Traditional PID, GWO-PID, SSA-PID, and ISSA-PID.

Figure 6 further confirms that all optimized controllers generate periodic position responses consistent with circular motion. GWO-PID and SSA-PID follow the expected sinusoidal trajectories more quickly, whereas ISSA-PID exhibits larger transient phase lag, particularly for the follower robot.

Figure 7 shows that all controllers maintain bounded velocities and achieve stable circular motion after convergence. However, angular velocity oscillations remain visible for all methods, suggesting that optimized PID gains improve tracking accuracy more effectively than motion smoothness under disturbance and noise conditions.

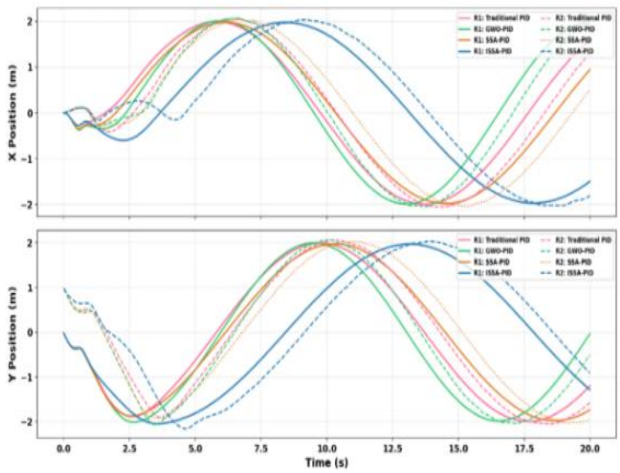

Figure 6: Time responses of the x- and y-positions for the leader and follower robots under different PID tuning methods.

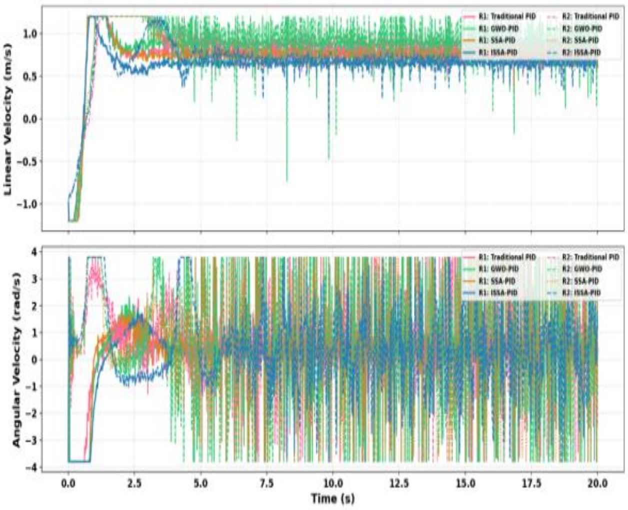

Figure 7: Linear and angular velocity profiles of the leader and follower robots under different PID tuning methods.

The path-tracking errors shown in Figure 8 indicate that the leader error decreases rapidly to a low level for all controllers. The follower error is larger during the transient phase, which is expected because the follower must track a moving reference while maintaining the desired formation distance. In the reported simulation case, the Traditional PID exhibits the largest transient peak and a prolonged settling time. Among the optimized controllers, ISSA-PID and SSA-PID demonstrate a rapid initial error reduction, although they experience minor transient oscillations before stabilizing. Notably, ISSA-PID achieves the fastest overall settling behavior, reaching the steady state much quicker than GWO-PID, which displays a smoother but significantly slower convergence

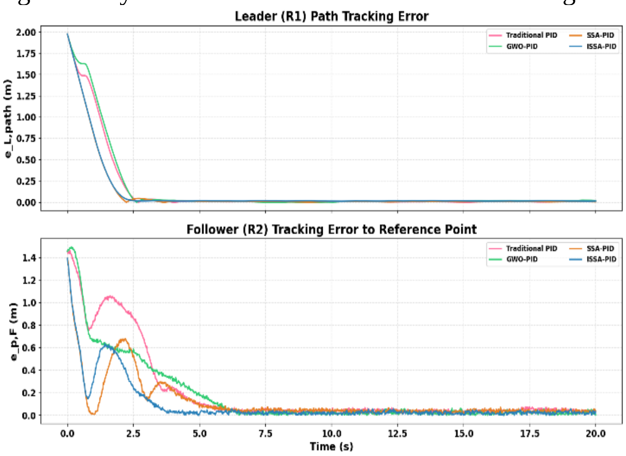

Figure 8: Path-tracking errors of the leader and follower robots under traditional PID, GWO-PID, SSA-PID, and ISSA-PID

Figure 9 illustrates the system snapshots at different time instants during the tracking process. At the early stage, noticeable deviations from the desired circular path and formation spacing are still observed, particularly for ISSA-PID. As time progresses, all methods gradually converge to stable circular motion, and the formation becomes more consistent. By the later stage of the simulation, the robot trajectories are close to the desired circle for all optimized controllers.

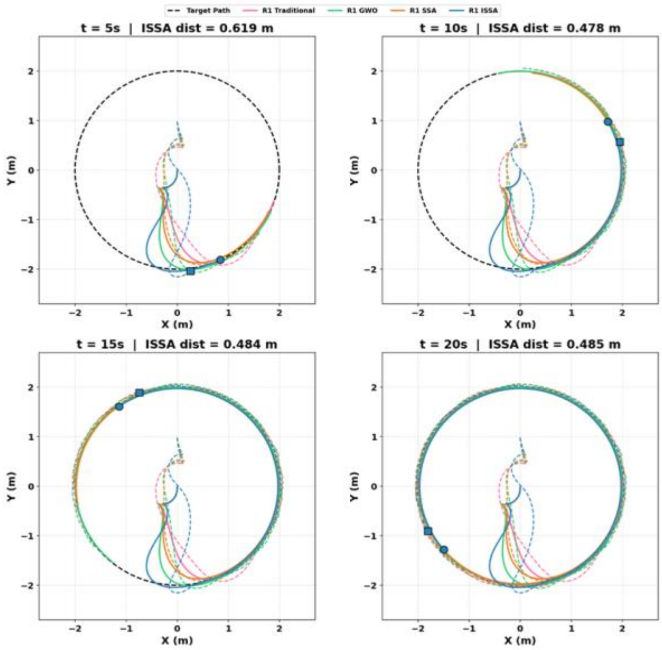

Figure 9: Time snapshots of leader–follower circular tracking performance.

Finally, Figure 10 summarizes the overall control quality in terms of tracking accuracy, transient response, formation maintenance, motion smoothness, and disturbance robustness. Based on the reported simulation results, GWO-PID and SSA-PID provide slightly better overall performance than ISSA-PID in this test case, particularly in transient response and follower tracking. ISSA-PID remains competitive and still improves the traditional PID baseline, but it does not consistently achieve the best behavior across all evaluated aspects.

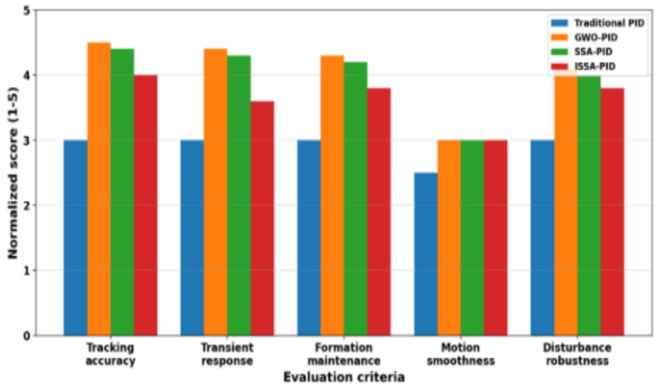

Figure 10: Control quality comparison

5.2 Discussion

Simulation results confirm that metaheuristic optimization effectively improves PID tuning for leader–follower path following, with all optimized controllers outperforming the manually tuned baseline under noisy and disturbed conditions. Statistical analysis over 20 runs shows that SSA and ISSA achieve lower mean objective values than GWO, with SSA providing the most stable performance and ISSA obtaining the lowest mean value with slightly higher variability. However, performance differences are insufficient to establish a definitive ranking, as all controllers achieve stable circular tracking and formation maintenance after the transient phase. Angular velocity oscillations remain present across all optimized controllers, suggesting that the proposed cost function prioritizes tracking accuracy over motion smoothness. Since these oscillations mainly arise from noise amplification in the PID derivative term, smoother rotational motion would require derivative filtering or additional smoothness-oriented design constraints. Overall, the present results should be interpreted as a comparative evaluation within a specific simulation setting rather than as a universal ranking of the three algorithms. Broader statistical testing and experimental validation are still needed before drawing stronger conclusions about the relative superiority of any single method.

References

[1] W. Chen, W. Chi, S. Ji, H. Ye, J. Liu, Y. Jia, J. Yu, and J. Cheng, "A survey of autonomous robots and multi-robot navigation: Perception, planning and collaboration," *Biomimetic Intelligence and Robotics*, vol. 5, no. 2, p. 100203, Jun. 2025, doi: 10.1016/j.birob.2024.100203.

[2] Z. Luo, B. Zhu, J. Zheng, and Z. Zheng, "Robust Distributed Model Predictive Control for Formation Tracking of Nonholonomic Vehicles," IEEE/CAA Journal of Automatica Sinica, vol. 11, no. 2, pp. 560–562, 2024, doi:10.1109/JAS.2023.123732.

[3] H. Xie, J. Zheng, Z. Sun, H. Wang, and R. Chai, "Finite-time tracking control for nonholonomic wheeled mobile robot using adaptive fast nonsingular terminal sliding mode," *Nonlinear Dyn.*, vol. 110, pp. 1437–1453, 2022, doi: 10.1007/s11071-022-07682-2.

[4] W. Zeng, T. Xiong, and C. Wang, "Optimization of Smooth Trajectories for Two-Wheel Differential Robots Under Kinematic Constraints Using Clothoid Curves," *Sensors*, vol. 25, no. 10, p. 3143, May 2025. https://doi.org/10.3390/s25103143

[5] J. G. Romero, E. Nuño, and C. I. Aldana, "Robust PID consensus-based formation control of nonholonomic mobile robots affected by disturbances," *Int. J. Control*, vol. 96, no. 3, pp. 791–799, 2023, doi:10.1080/00207179.2021.2015541.

[6] J. Tang, S. Wu, B. Lan, Y. Dong, Y. Jin, G. Tian, W.-A. Zhang, and L. Shi, "GMPC: Geometric Model Predictive Control for Wheeled Mobile Robot Trajectory Tracking," IEEE Robotics and Automation Letters, vol. 9, no. 5, pp. 4822–4829, 2024, doi: 10.1109/LRA.2024.3381088.

[7] T. L. Nguyen, X. S. Mai, and P. N. Dao, "A robust distributed model predictive control strategy for Leader-Follower formation control of multiple perturbed wheeled mobile robotics," *Eur. J. Control*, vol. 81, p. 101160, Jan. 2025, doi: 10.1016/j.ejcon.2024.101160.

[8] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[9] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Syst. Appl.*, vol. 166, p. 113917, 2021, doi: 10.1016/j.eswa.2020.113917.

[10] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, 2020, doi: 10.1080/21642583.2019.1708830.

[11] Y. He and M. Wang, "An improved chaos sparrow search algorithm for UAV path planning," *Sci. Rep.*, vol. 14, no. 1, p. 366, 2024, doi: 10.1038/s41598-023-50484-8.

[12] X. Jiang, W. Wang, Y. Guo, and S. Liao, "A Multi-Strategy Crazy Sparrow Search Algorithm for the Global Optimization Problem," *Electronics*, vol. 12, no. 18, p. 3967, 2023, doi: 10.3390/electronics12183967.

[13] B. Morales-Castañeda, D. Zaldívar, E. Cuevas, F. Fausto, and A. Rodríguez, "A better balance in metaheuristic algorithms: Does it exist?" *Swarm Evol. Comput.*, vol. 54, p. 100671, 2020, doi: 10.1016/j.swevo.2020.100671.

6. Conclusion

This paper evaluated GWO, SSA, and ISSA for PID tuning in leader-follower mobile robot circular path following under disturbances. Statistical analysis demonstrated that SSA and ISSA outperform GWO; SSA provides the highest stability across repeated runs, while ISSA yields the optimal mean objective value. Although stable tracking is successfully achieved, persistent angular velocity oscillations indicate the inherent limitations of linear PID control. To enhance motion smoothness and robustness against mismatched disturbances, our future work will focus on integrating Sliding Mode Control architectures with Reinforcement Learning algorithms.

[14] Y. Xu, B. Sang, and Y. Zhang, "Application of Improved Sparrow Search Algorithm to Path Planning of Mobile Robots," *Biomimetics*, vol. 9, no. 6, p. 351, 2024, doi: 10.3390/biomimetics9060351.

[15] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems," *Heliyon*, vol. 8, no. 5, p. e09399, 2022, doi: 10.1016/j.heliyon.2022.e09399.

[16] E. Bayona, J. E. Sierra-García, and M. Santos, "Comparative Analysis of Metaheuristic Optimization Methods for Trajectory Generation of Automated Guided Vehicles," *Electronics*, vol. 13, no. 4, p. 728, 2024, doi: 10.3390/electronics13040728.

[17] J. Hou, W. Jiang, Z. Luo, L. Yang, X. Hu, and B. Guo, "Dynamic Path Planning for Mobile Robots by Integrating Improved Sparrow Search Algorithm and Dynamic Window Approach," *Actuators*, vol. 13, no. 1, p. 24, 2024, doi: 10.3390/act13010024.

[18] S. Guillén Ruiz, L. V. Calderita, A. Hidalgo-Paniagua, and J. P. Bandera Rubio, "Measuring Smoothness as a Factor for Efficient and Socially Accepted Robot Motion," *Sensors*, vol. 20, no. 23, p. 6822, 2020, doi: 10.3390/s20236822.