International Journal of Dynamics and Control (2026) 14:36
https://doi.org/10.1007/s40435-025-01967-1

Optimal real PIDD^2 controller design for vehicle cruise control via Lévy flight-based White Shark Optimizer with Nelder–Mead algorithm

Hasan Başak[1]

Received: 2 September 2025 / Revised: 6 November 2025 / Accepted: 29 November 2025 / Published online: 26 December 2025

Abstract

Vehicle cruise control systems have become essential components of modern automobiles, offering significant benefits in driving comfort, fuel efficiency, and road safety. An effective control strategy is crucial for ensuring reliable performance. In this paper, a real PIDD^2 (proportional–integral–derivative plus second-order derivative) controller is proposed to address the challenges associated with achieving efficient and robust cruise control. To optimally tune the real PIDD^2 controller parameters, we develop a novel metaheuristic approach that enhances the white shark optimization (WSO) algorithm using Lévy flight and the Nelder–Mead method. While Lévy flight improves global exploration, the Nelder–Mead method strengthens local exploitation, resulting in the proposed L-WSONM algorithm. To validate its robustness and effectiveness, L-WSONM is extensively tested on complex optimization problems from the CEC2020 benchmark suite and compared with several high-performance algorithms, including gray wolf, pufferfish, golden jackal, reptile search, white shark, tunicate swarm, prairie dog, and seagull optimizers. The results show that L-WSONM consistently delivers superior solution quality and faster convergence rates, with the exception of functions F2 and F9. Finally, L-WSONM is applied to fine-tune the parameters of the real PIDD^2 controller for vehicle cruise control. Simulation results demonstrate that the L-WSONM-based real PIDD^2 controller achieves zero overshoot, a rise time of 0.463 s, and a settling time of 0.692 s, outperforming the original WSO and other recent approaches from the literature.

Keywords Cruise control system · White Shark Optimizer · Lévy flight · Nelder–Mead method · Real PIDD^2 controller design

1 Introduction

Vehicle cruise control systems have emerged as essential features in modern automobiles produced in recent years. They significantly contribute to enhancing driving comfort, reducing fuel consumption, and improving overall road safety [1]. Cruise control systems are intended to preserve a vehicle's speed at a desired setpoint, thereby reducing the driver's workload related to throttle control. This enables the driver to focus more effectively on other critical aspects of vehicle operation [2, 3]. Hence, developing effective methods to design an optimal controller for cruise control systems is of critical importance [4].

✉ Hasan Başak
hasanbasak@artvin.edu.tr

1 Electrical-Electronics Engineering, Artvin Çoruh University, 08100 Artvin, Turkey

In the literature, numerous control strategies have been proposed for vehicle cruise control systems. These include model predictive control [5, 6], state-space [7], adaptive H_∞ control [8], adaptive control [9], fuzzy logic control [10], and deep reinforcement learning-based approach [11]. While many of these methods can deliver satisfactory performance, their design and implementation processes are often complex and time-consuming, and they may not consistently guarantee optimal results. Consequently, the identification and development of an effective controller remains a challenging task that necessitates further investigation.

Metaheuristic algorithms have emerged as powerful tools for solving complex optimization problems. To cite a few, recent studies on optimization methods for control system design and parameter estimation are reported in [12–15]. In the context of vehicle cruise control systems, metaheuristic algorithms offer significant potential to improve transient response and strengthen system stability. In the literature, PID controllers are often preferred over more advanced control methods due to their simplicity and ease of implementation [16–19]. For improved performance, a FOPID controller may be utilized, offering increased flexibility in tuning due to its fractional-order components [20]. Nevertheless, the inclusion of fractional terms increases the computational complexity [21]. Compared to PID and FOPID controllers, the real PID plus second-order derivative (RPIDD^2) controller has demonstrated improved steady-state accuracy and stability in simulation studies [22, 23], making it a promising option for the reliable operation of vehicle cruise control systems. Table 1 summarizes a comparative study of various optimization-based cruise control systems and reports their corresponding performance metrics, including overshoot percentage (OS%), rise time (t_r (s)), and settling time (t_s (s)). Although optimization frameworks have advanced, there is still substantial room for enhancement. Notably, the no free lunch theorem [24] states that no algorithm is better than any other algorithm for every problem and that the most effective method depends on the specific characteristics of each problem.

The White Shark Optimizer (WSO) [25], a recent algorithm, mimics the hunting behavior of white sharks in their natural ocean environment. These sharks explore their environment using a specialized sense of hearing that extends along both sides of their bodies. Additionally, they possess an exceptional sense of smell that intensifies as they approach their prey, enabling precise localization. In recent research, several modified versions of the WSO have been proposed to address various engineering and optimization problems. These include an enhanced WSO for solving the optimal power flow problem [26], a chaotic WSO tailored for engineering applications [27], a modified WSO designed for hybrid renewable energy systems [28], and an improved WSO applied to photovoltaic systems under partial shading conditions [29]. Additional developments feature an opposition-based learning WSO for pavement maintenance planning [30], a boosting WSO for addressing cloud scheduling problems [31], and a hybrid WSO combined with the whale optimization algorithm for parameter identification in Li-Ion battery models [32].

In this study, a new tuning strategy is proposed for the cruise control system governed by the RPIDD^2 control strategy through the development of an innovative metaheuristic algorithm. Building upon the original WSO, the approach is enhanced through the integration of the Lévy Flight mechanism and the Nelder–Mead strategy. This modified version, referred to as L-WSONM, significantly improves the original WSO algorithm by achieving a well-balanced trade-off between exploration and exploitation. The algorithm is designed to achieve more accurate and efficient identification of optimal RPIDD^2 controller parameters.

A comprehensive performance evaluation is carried out, including statistical metrics, Wilcoxon's nonparametric test, convergence behavior, and transient response analysis. The findings clearly indicate that the L-WSONM algorithm significantly outperforms the original WSO. A comparative evaluation is also undertaken to assess the performance of the proposed approach relative to traditional PID, fractional-order PID, and RPIDD^2 control schemes. To validate the effectiveness of the developed controller, detailed comparisons are conducted against PID controllers optimized through various advanced metaheuristic algorithms. These include the red panda optimization algorithm (RPO) [33], the arithmetic optimization incorporating opposition-based learning and local refinement (ObAOANM) [34], the hybrid arithmetic optimization algorithm with nelder–mead (AOANM) [35], Lévy flight-enhanced reptile search algorithm (RSALF) [36] and manta ray-inspired foraging optimization (MRFO) [35]. In addition, the proposed method is benchmarked against FOPID controllers tuned using several state-of-the-art optimization algorithms, such as Harris hawks optimization (HHO) [20], the artificial hummingbird algorithm (AHA) [37], the genetic algorithm (GA) [38], the boosted weighted mean of vectors (b-INFO) [39] and the modified artificial hummingbird algorithm (mAHA) [37]. To provide further evidence of the proposed method's superiority, comparative analyses are performed against RPIDD^2 controllers optimized by widely recognized algorithms, including the particle swarm optimization (PSO) [22], the salp-inspired swarm intelligence algorithm (SSA) [22], and the Aquila Optimizer enhanced with a modified oppositional chaotic local search strategy (CmOBLAO) [22], Runge–Kutta optimizer (RUN) [3] and its improved Runge Kutta optimizer (IRUN) [3].

Through benchmarking against established optimization techniques, the proposed approach consistently demonstrates superior effectiveness in improving vehicle cruise control system performance. This comprehensive investigation emphasizes enhancing time-domain response and system stability by leveraging the capabilities of the L-WSONM algorithm. The main contributions of this study can be summarized as follows:

- An enhanced optimization method, termed the L-WSONM optimizer, is proposed as an improved variant of the WSO algorithm by integrating the Lévy Flight and Nelder–Mead strategies.
- Validation of the proposed algorithm's effectiveness is conducted using the CEC2020 benchmark problems. Its performance is compared against several advanced algorithms, including gray wolf optimization (GWO) [40], pufferfish optimization algorithm (POA) [41], golden jackal optimization (GJO) [42], reptile search algorithm (RSA) [43], white shark optimizer (WSO) [25], tunicate swarm algorithm (TSA) [44], genetic algorithm (GA) [45], prairie dog optimization algorithm (PDO) [46], and

Table 1 Summary of recent studies on optimization-based controller design of vehicle cruise control systems

Ref	Controller	Optimizer	t_r (s)	t_s (s)	OS (%)	Advantages	Disadvantages
[16]	PID	ALO	0.8094	1.732	0.612	Outperforms state–space, fuzzy logic, GA-tuned, and conventional PID controllers	Exhibits relatively high transient characteristics
[17]	PID	PSO	0.2818	1.52	17.77	Substantially improves the transient response compared to Ziegler–Nichols and fuzzy controllers	Still shows nonzero overshoot and relatively high settling time
[18]	PID	ASO–NM	0.8052	1.2622	0.3541	Achieving a well-balanced exploration–exploitation behavior	Some transient characteristics remain relatively high
[19]	PID	GA	0.851	2.59	4.51	Achieves faster rise time	Shows noticeable overshoot and higher settling time
[20]	FOPID	HHO	0.7019	1.1034	0.0642	Outperforms ALO-PID, and GA-PID across all key performance metrics	FOPID structure increases computational cost
[22]	RPIDD^2	CmOBLAO	0.6540	0.9985	0	First implementation of a real PIDD^2 controller	Transient improvements are moderate
[33]	PID	RPO	0.6921	1.0808	0.1206	Enhances transient performance	Improvements are not significantly good
[34]	PID	ObAOANM	0.7457	1.1729	0.0108	Achieves a well-balanced exploration–exploitation mechanism	Transient performance improvements remain modest
[36]	PID	RSALF	0.6925	1.0723	0.3544	Better transient performance and superior results compared to existing methods	The PID structure limits achievable performance compared to more advanced control designs
[37]	FOPID	mAHA	0.6630	1.0427	0	Improved safety and reliability due to smoother and more accurate tracking	Exhibits a relatively high settling time
[38]	FOPID	GA	0.80665	1.8292	2.0597	Improved stability and accuracy	Exhibits a relatively high settling time
[39]	PI^{λ}DN-D^2N^2	b-INFO	0.6549	0.9893	0	Achieves the most efficient cruise control performance among numerous recent methods	Improvements in transient characteristics may be marginal in certain performance metrics

seagull optimization algorithm (SOA) [47]. Statistical metrics and nonparametric tests confirm that the proposed algorithm outperforms these competitors in both convergence speed and solution quality.

- The proposed L-WSONM optimizer is, for the first time, applied to enhance the efficiency of the RPIDD^2 controller in the context of vehicle cruise control.
- The effectiveness of the L-WSONM-based RPIDD^2 controller is extensively verified through comparative analysis with PID, fractional-order PID, and conventional RPIDD^2 controllers, with an emphasis on detailed performance evaluation. Comprehensive analyses are conducted, including statistical performance metrics, convergence behavior, stability assessments, and comparisons with established methods. These comprehensive evaluations confirm the robustness and superior performance of the proposed method across various parameter changes.

The remainder of this paper is organized as follows: Sect. 2 presents the structural frameworks of both the original WSO algorithm and the proposed L-WSONM approach. The effectiveness of the proposed algorithm is validated in Sect. 3 through the CEC2020 benchmark problems. Section 4 details the mathematical model of the cruise control system and the RPIDD^2 controller, including their integration for effective speed control. In Sect. 6, the proposed algorithm's effectiveness is thoroughly evaluated through benchmark comparisons with existing methods. Finally, Sect. 7 concludes the paper.

2 L-WSONM algorithm

2.1 White Shark Optimizer

The White Shark Optimizer (WSO) was proposed in [25] which was mostly inspired by characterizing the behavior of white sharks while foraging. Inspiration for the algorithm stems from the behavioral traits of great white sharks, especially their highly developed hearing and sense of smell, which aid in movement and prey detection. The mathematical modeling of these behaviors of the WSO is initialized as follows:

$$\begin{bmatrix} Z_1 \\ \vdots \\ Z_i \\ \vdots \\ Z_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} z_{1,1} & \dots & z_{1,j} & \dots & z_{1,d} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ z_{i,1} & \dots & z_{i,j} & \dots & z_{i,d} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ z_{N,1} & \dots & z_{N,j} & \dots & z_{N,d} \end{bmatrix}_{N\times d} \tag{1}$$

The search space is populated with initial candidate solutions as follows:

$$z_{ij} = lb_j + r \times (ub_j - lb_j) \tag{2}$$

where z_{ij} denotes the initial position of the ith white shark in the jth dimension, N is the population of the algorithm and d is the number of decision variable, ub_j and lb_j represent the upper and lower bounds of the search space in the jth dimension, respectively, and r is a randomly generated number within the interval [0, 1].

White sharks utilize a combination of auditory, olfactory, and visual senses to detect and track their prey. Additionally, they can perceive pressure changes in the surrounding water caused by the prey's wave movements, which helps guide them toward their target. The speed of movement of the white shark is given as follows:

$$\begin{aligned} v_i^{t+1} &= \mu(v_i^t + p_1(z_{\text{gbest}}^t - z_i^t) \times c_1 \\ &\quad + p_2(z_{\text{best}}^{v_i^t} - z_i^t) \times c_2) \end{aligned} \tag{3}$$

where $i = 1, 2, \dots, n$ is the index of a white shark in a population of size n. v_i^{t+1} denotes the updated velocity vector of the ith white shark in the $(t+1)$th iteration, v_i^t is its velocity in the tth iteration, z_{gbest}^t is the global best position found by any white shark up to iteration t, z_i^t is the current position of the ith white shark in the tth iteration, c_1 and c_2 are random variables uniformly distributed over the interval [0, 1], $z_{\text{best}}^{v_i^t}$ represents the best known position vector associated with the ith shark in the swarm during the tth iteration and the index vector of the sharks that have reached the optimal position is

$$v = n \times rand(1, n) + 1 \tag{4}$$

The parameters p_1 and p_2 represent the forces of the white sharks, which influence the effects of z_{gbest}^t and $z_{best}^{v_i}$ on z_i^t; they can be calculated as follows:

$$p_1 = p_{\max} + (p_{\max} - p_{\min}) \cdot e^{-\left(\frac{4t}{T}\right)^2} \tag{5}$$

$$p_2 = p_{\min} + (p_{\max} - p_{\min}) \cdot e^{-\left(\frac{4t}{T}\right)^2} \tag{6}$$

where t denotes the current iteration, and T represents the total number of iterations. $p_{\min}$ and $p_{\max}$ represent the lower and upper velocity limits, respectively, set to enhance the movement efficiency of great white sharks, with $p_{\min} = 0.5$, $p_{\max} = 1.5$ [25]. The constriction factor is

$$\mu = \frac{2}{2 - \tau - \sqrt{\tau^2 - 4\tau}} \tag{7}$$

where τ is the acceleration parameter and equals 4.125 [25]. Great white sharks spend most of their time searching for prey where an optimal or suboptimal prey can be located. Sharks adjust their positions based on the movement sound or the scents of prey. If the prey moves from its position, sharks continue tracking by scent, often exploring randomly–similar to fish schools searching for food. In this context, the position update mechanism given in Eq. (8) is employed to model the movement behavior of white sharks as they approach their prey.

$$z_i^{t+1} = \begin{cases} z_i^t \cdot \neg \oplus z_o + ub \cdot a + lb \cdot b, & \text{if rand} < mv \\ z_i^t + \dfrac{v_i^t}{f}, & \text{if rand} \geq mv \end{cases} \quad (8)$$

where z_i^{t+1} indicates the updated position vector of the ith white shark at iteration $t+1$, $\neg$ represents the negation operator and $\oplus$ is the bit-wise xor operation. a and b are one-dimensional binary vectors, defined as follows.

$$a = \text{sgn}(z_i^t - u) > 0 \quad (9)$$

$$b = \text{sgn}(z_i^t - l) < 0 \quad (10)$$

z_o denotes a logical vector defined as

$$z_o = \oplus(a, b) \quad (11)$$

f denotes the frequency of the white shark's wavy motion and is given as flows.

$$f = f_{\min} + \frac{f_{\max} - f_{\min}}{f_{\max} + f_{\min}} \quad (12)$$

and mv denotes the movement force that grows with the number of iterations as the white shark gets closer to its prey, and is formulated by

$$mv = \left(\frac{1}{a_0 + e^{\;(\frac{T}{2} - t)/a_1}} \right) \quad (13)$$

where a_0 and a_1 are parameters that control the balance between exploration and exploitation. The parameter mv represents the intensity of the white shark's auditory and olfactory senses, which grow progressively with the number of iterations.

Great white sharks are capable of maintaining their orientation toward the best shark which is near the prey, as described in Eq. (14)

$$\vec{z}_i^{\,t+1} = z_{gbest}^t + r_1 \times \vec{D}_z \times \text{sgn}(r_2 - 0.5), \quad \text{if } r_3 < S_s \quad (14)$$

where $\vec{z}_i^{\,t+1}$ is the updated position of the ith white shark relative to the position of the prey. $\vec{D}_z$ represents the distance between the white shark and the prey (i.e., food source), as defined in Eq.(15). The parameter S_s, defined in Eq. (16), reflects the intensity of the white shark's sense of smell and sight when following others that are near the optimal prey.

$$\vec{D}_z = |\text{rand} \times \left(z_{\text{gbest}}^t - z_i^t \right)| \quad (15)$$

$$S_s = 1 - e^{\frac{-a_2 t}{T}} \quad (16)$$

where a_2 is positive constant.

In simulating the white shark school behavior, the two fittest solutions are maintained, and the rest of the population adjusts their positions relative to these optimal references. The positions of all other sharks are updated using Eq. (17).

$$z_i^{t+1} = \frac{z_i^t + \vec{z}_i^{\,t+1}}{2 \times rand} \quad (17)$$

The subsequent subsections delve into the foundational principles of the Lévy flight mechanism and the Nelder–Mead simplex technique, culminating in the formulation of the proposed L-RSANM optimization algorithm.

2.2 Lévy flight

The Lévy flight operator is effective for generating random motion within a defined design space, thereby enhancing the diversification capability of a metaheuristic algorithm [48]. Consequently, the Lévy flight mechanism has been extensively adopted in a range of metaheuristic optimization strategies. Notable recent implementations include its integration with the backtracking search algorithm [49] and the grasshopper optimization algorithm [50] for photovoltaic model parameter identification, as well as with the reptile search algorithm [51] for control system design in power engineering applications.

The Lévy flight mechanism is defined by a stochastic process with a non-Gaussian distribution, typically characterized by $L(s) \sim |s|^{-1-\beta}$ [52] and can be represented as follows: Here, s denotes the step length, μ is the transmission parameter, and λ controls the scale of the distribution. The Lévy flight distribution is commonly expressed through its Fourier transform as follows:

$$F(k) = e^{-\alpha |k|^{\beta}} \quad (18)$$

in which α denotes the scaling factor and β is the distribution index constrained to the interval (0, 2]. The Lévy step is defined as follows:

$$s = \frac{u}{|v|^{1/\beta}} \quad (19)$$

where both u and v follow a Gaussian distribution and they are given as

$$u \sim \mathcal{N}(0, \sigma_u^2), \quad v \sim \mathcal{N}(0, 1) \quad (20)$$

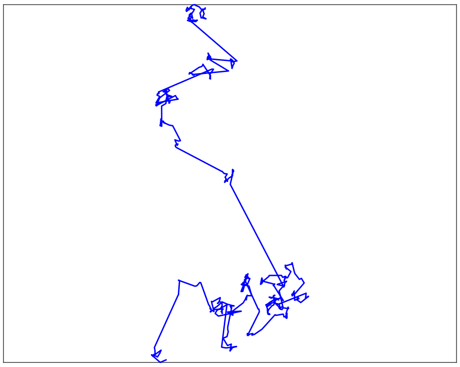

Fig. 1 Two dimensional Lévy flights

where σ_u denotes standard deviation which is given by

$$\sigma_u = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta} \tag{21}$$

Figure 1 illustrates a two-dimensional Lévy flight trajectory.

2.3 Nelder–Mead algorithm

The Nelder-Mead (NM) algorithm [53] is a direct search technique that has been applied to many problems with encouraging results. It is usually used for optimizing locally. In the literature, a number of works have showed that how well the NM approach performs when it comes to improving the performance of metaheuristic structures. Among these improved metaheuristic optimization algorithms are: parameter extraction of photovoltaic systems utilizing moth-flame optimization [54]; ecosystem-inspired optimization [55] for implementing a PID-controlled buck converter system; the reptile-inspired search algorithm [56] with refined search strategy in power system applications; and the Lévy flight-inspired algorithm [57] for Speed regulation of a DC motor.

In NM algorithm, a d-dimensional simplex is formed by $d+1$ vertices $(z_1, z_2, ...z_d, z_{d+1})$. A sequence of simplexes is generated to iteratively approximate the optimal point of the objective function, $f(z)$. Each iteration involves rearranging the simplex's vertices to produce the following relationship: $f(z_1) \leq f(z_2) \leq \cdots \leq f(z_d+1)$, where z_1 and z_{d+1} represent the best and the worst vertices, respectively. NM algorithm consists of following stages: (i) sorting, (ii) reflection, (iii) expansion, (iii) contraction, and (v) shrinkage. The objective function is sorted with respect to its values as $f(z_1) \leq f(z_2) \leq \cdots \leq f(z_{d+1})$. Then, the reflection point is given as

$$z_r = \bar{z} + \rho(\bar{z} - z_{d+1}) \tag{22}$$

where $\bar{z}$ is a centroid point that is

$$\bar{z} = \frac{1}{d} \sum_{i=1}^{d} z_i \tag{23}$$

Once the reflection point is known, then $f(z_r)$ is computed. z_{d+1} is replaced by z_r for the condition of $f(z_1) \leq f(z_r) < f(z_d)$. If $f(z_r) \leq f(z_1)$, then the expansion point is obtained as follows

$$z_e = \bar{z} + \gamma(z_r - \bar{z}) \tag{24}$$

if $f(z_e) < f(z_r)$, z_e replaces the worst value, on the other hand x_{d+1} is equal to z_r. Next, the outside contraction point is calculated as follows for the case $f(z_d) \leq f(z_r) < f(z_{d+1})$.

$$z_{oc} = \bar{z} + \beta(z_r - \bar{z}) \tag{25}$$

In this step, z_{oc} takes the place of z_{d+1} if $f(z_{oc}) \leq f(z_r)$. Alternatively, the shrinkage phase is executed. For $f(z_{d+1}) \leq f(z_r)$, an inside contraction point is computed as

$$z_{ic} = \bar{z} + \beta(z_{d+1} - \bar{z}) \tag{26}$$

in the next step, z_{ic} takes the place of $z_{d+1)}$ if $f(z_{ic}) < f(z_{d+1})$. Similar to the outside contraction scenario, a shrinkage operation is carried out at this step if $f(z_{ic}) > f(z_{d+1})$.. The shrinkage progress is described as follows

$$z_i = z_1 + \delta(z_i - z_1) \tag{27}$$

The latter is applied to form new points in the shrinking progress. Figure 2 illustrates the step-by-step flow of the NM simplex algorithm.

2.4 Structure of L-WSONM algorithm

The proposed L-WSONM algorithm integrates the Lévy flight operator with the NM method to improve the main structure of the WSO. The following equation governs the generation of a new search agent, which is updated through the application of Lévy flight after each iteration [56].

$$z_i^{LF} = z_i + [2 \cdot r - 1] \cdot \text{Levy}(\beta) \cdot (z_{\text{best}} - z_i) \tag{28}$$

Here, z_i^{LF} represents the new solution generated by the Lévy flight operation, and z_i denotes the present solution, r is a uniformly distributed random number in [0, 1], and z_{best} signifies the best solution encountered so far. Both the candidate solutions z_i and the Lévy flight solutions z_i^{LF} are evaluated collectively. From the combined set $z_i \cup z_i^{LF}$, the top N solutions are selected. Thus, at each iteration, improved solutions

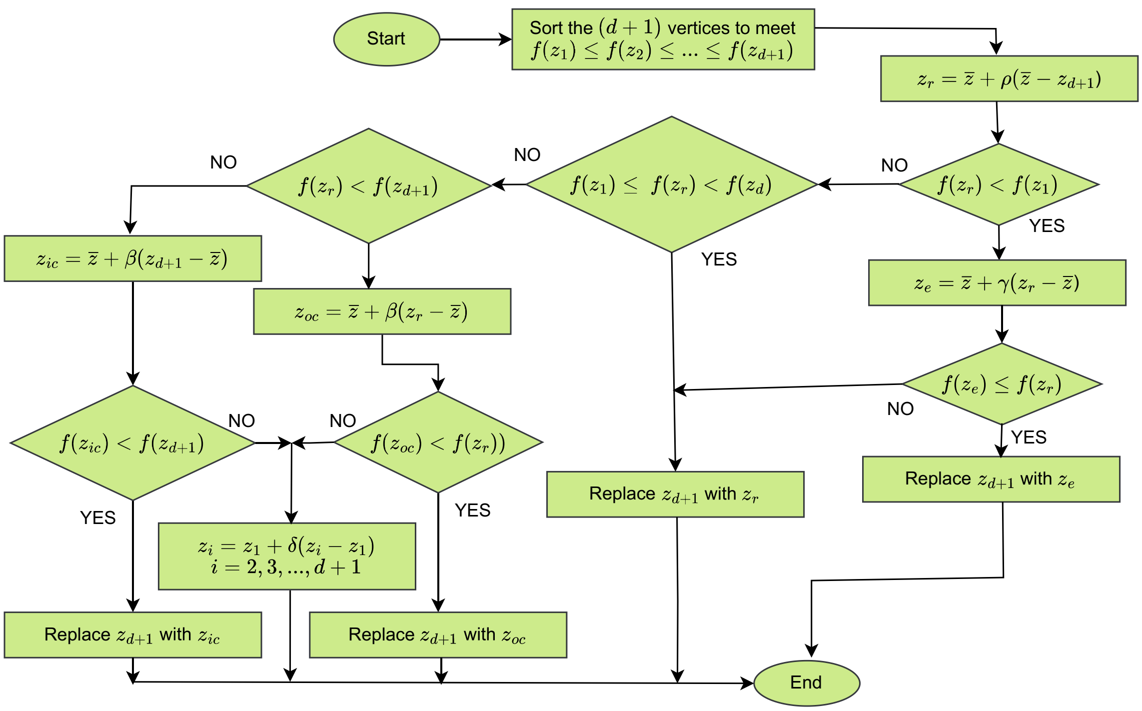

Fig. 2 Flowchart of the Nelder-Mead algorithm [55]

are retained and refined for subsequent updates, whereas inferior ones are systematically discarded.

Exploitation is performed using the NM optimization method, which is invoked at a trigger interval of α iterations. Additional executions of the NM step continue to produce performance improvements. As demonstrated in [57], the NM procedure is applied at every iteration, despite the associated increase in computational cost. Simulation results confirm that this application of NM significantly enhances performance in our approach. This method maintains an ideal balance between diversification and intensification in the search process. The operational flow of the developed L-WSONM algorithm is comprehensively illustrated in Fig. 3.

2.5 Computational complexity

Computational complexity can be estimated by examining the structural components and operations of the algorithm. The time complexity of WSO is given in [25] as $\mathcal{O}(TcN + TNd)$ where T is the maximum number of iterations, c denotes the cost of a single fitness evaluation, N is the population size, and d is the problem dimension. The Lévy–flight update introduces an additional per-agent vector operation of cost $O(d)$ together with an additional fitness evaluation of cost c. Therefore, its per-iteration complexity is $O(Nd+Nc)$. Including this term, the overall time complexity becomes $O(\text{WSO} + \text{Lévy}) = O(TcN + TNd + TNc + TNd)$.

The Nelder–Mead (NM) local search is invoked every α iterations. In the proposed algorithm, one NM call requires $O(dT)$ operations; therefore, the total NM cost over the run is $O\left(\frac{T}{\alpha}dT\right) = O\left(\frac{T^2}{\alpha}d\right)$. Assuming the fitness evaluation cost satisfies $c = O(d)$, the overall computational complexity of the proposed L–WSONM algorithm becomes $O(TNd) + O\left(\frac{T^2}{\alpha}d\right)$. Although L-WSONM has a higher computational cost than its individual component algorithms, this cost is justified by its faster convergence behavior and its enhanced ability to escape local minima.

3 Experimental results on CEC2020 benchmark functions

In this section, we utilize ten modern CEC2020 benchmark test functions from [58] to conduct a preliminary evaluation of the proposed L-WSONM algorithm. This serves as an effective reference for evaluating the performance of the algorithms. The details of these test functions are provided in Table 2.

For the purpose of statistical comparison with published works in the literature, we employed gray wolf optimization (GWO) [40], pufferfish optimization algorithm (POA) [41], golden jackal optimization (GJO) [42] reptile search algorithm (RSA) [43], White Shark Optimizer (WSO) [25], Tunicate Swarm Algorithm (TSA) [44], Genetic Algorithm

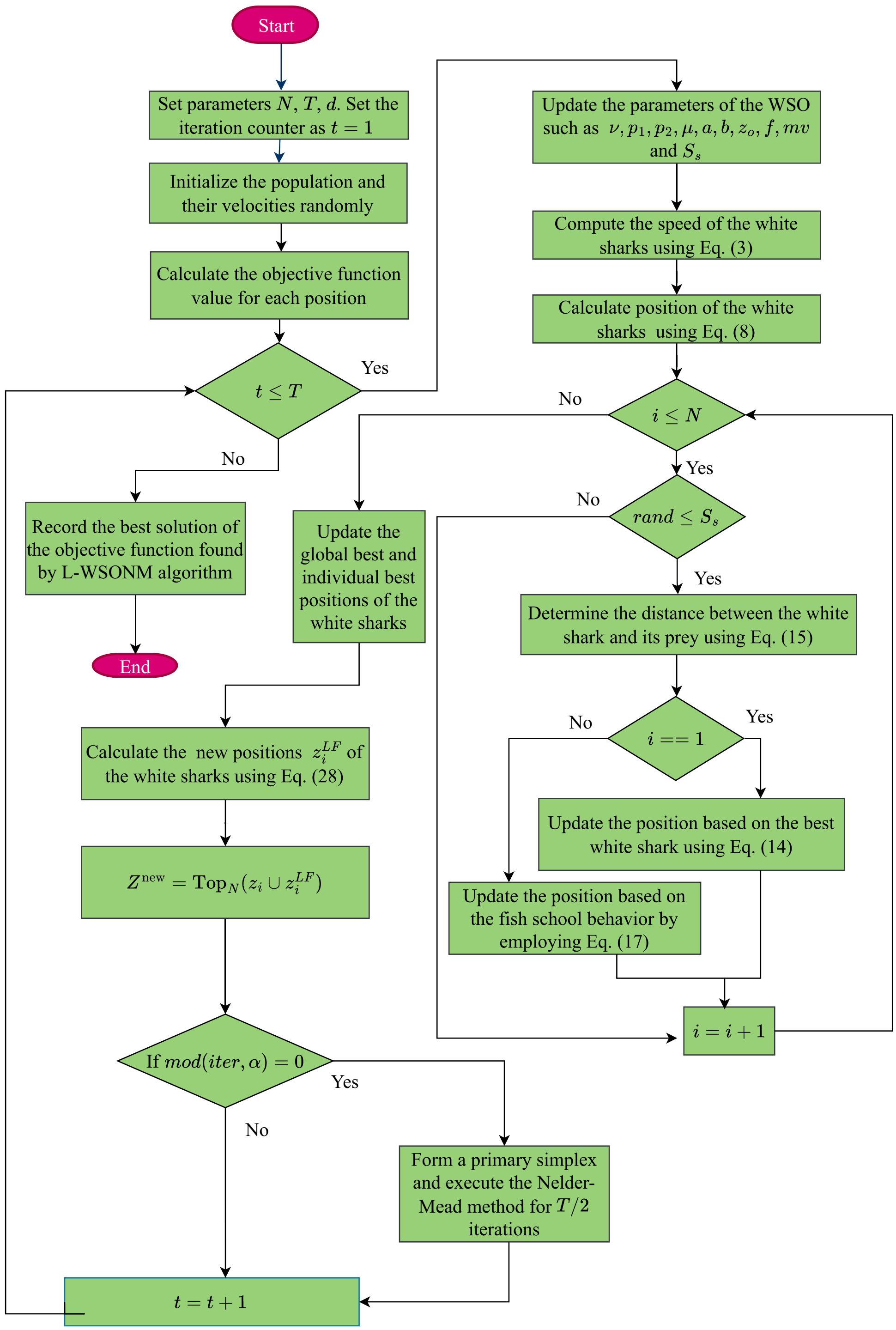

Fig. 3 Flowchart of the developed L-WSONM approach

(GA) [45] Prairie dog optimization algorithm (PDO) [46] and Seagull Optimization Algorithm (SOA) [47].The parameter settings of all algorithms were kept unchanged for fair comparison, ensuring that each competitor was configured exactly as specified in its original publication.

3.1 Discussion of the results on CEC2020 benchmark functions

The optimization of the CEC2020 benchmark test functions was performed using a population size of 30 and a maximum of 500 iterations. Moreover, the Lévy flight parameter

Table 2 Overview of the benchmark functions in CEC2020

Function	Definition	Dimension	Search Range	$F_{\min}$
F1	Shifted and rotated bent cigar function	10	[−100, 100]	100
F2	Shifted and rotated Schwefel's function	10	[−100, 100]	1100
F3	Shifted and rotated Lunacek bi-Rastrigin function	10	[−100, 100]	700
F4	Expanded Rosenbrock's plus Griewank's function	10	[−100, 100]	1900
F5	Hybrid function 1 ($N = 3$)	10	[−100, 100]	1700
F6	Hybrid function 2 ($N = 4$)	10	[−100, 100]	1600
F7	Hybrid function 3 ($N = 5$)	10	[−100, 100]	2100
F8	Composition function 1 ($N = 3$)	10	[−100, 100]	2200
F9	Composition function 2 ($N = 4$)	10	[−100, 100]	2400
F10	Composition function 3 ($N = 5$)	10	[−100, 100]	2500

β employed in our proposed algorithm was set to 1.5, consistent with the setting reported in [56]. Each algorithm was executed thirty times to ensure a fair and statistically reliable comparison.

First, we analyze the effect of different Nelder–Mead trigger intervals, and the corresponding results are reported in Table 3. According to the table, $\alpha = 10$ delivers the best overall performance, achieving the lowest sum and mean ranks among all configurations. This indicates that more frequent activation of the Nelder–Mead local search enhances exploitation and leads to improved convergence accuracy across the benchmark functions. In contrast, the configurations with $\alpha = 25$ and $\alpha = 50$ perform similarly but remain clearly inferior, suggesting that larger trigger intervals delay local refinement and reduce solution quality. Overall, $\alpha = 10$ provides the most effective balance between exploration and exploitation for the proposed L-WSONM.

To further evaluate the contribution of each component, ablation experiments were conducted, and the results are presented in Table 4. The findings clearly demonstrate that both the Nelder–Mead and Lévy flight mechanisms contribute meaningful performance enhancements to the baseline WSO. The hybrid variants–L-WSONM($\alpha = 10$) and WSONM($\alpha = 10$)–achieve the best overall results, with identical sum and mean ranks across the CEC2020 benchmark suite, indicating strong and consistent optimization capability. Among them, L-WSONM($\alpha = 10$) provides the most balanced performance in terms of accuracy and stability, particularly on more complex functions where the integration of global and local search becomes crucial.

In terms of runtime, the inclusion of the Nelder–Mead and Lévy flight components inevitably increases the computational cost compared to the baseline WSO, since additional local search steps and stochastic perturbations are executed. Although this overhead is relatively modest, it represents the primary drawback of L-WSONM.

For comparison with published works in the literature, the average convergence behaviors across thirty runs are shown in Fig. 4. Except for functions F2 and F9, the proposed L-WSONM demonstrates the fastest convergence rate among all competing algorithms, indicating its effectiveness and capability in achieving superior results. The box plots in Fig. 5 depict the data distributions from the 30 independent runs, where a narrower box plot represents greater consistency. For most benchmark functions, the proposed L-WSONM yields narrower box plots and lower objective values compared to other algorithms, highlighting its robustness and superior performance.

Table 5 presents the results of the statistical metrics including average (mean), standard deviation(std), minimum (Best) and Maximum (Worst). The mathematical description of these metrics is provided as follows:

$$\text{Average} = \frac{1}{E_n}\sum_{i=1}^{E_n} f_{opt,i} \tag{29}$$

$$\text{Std} = \sqrt{\frac{1}{E_n - 1}\sum_{i=1}^{E_n} (f_{opt,i} - \text{mean})^2} \tag{30}$$

$$\text{Best} = \min_{i=1}^{E_n}\{f_{opt,i}\} \tag{31}$$

$$\text{Worst} = \max_{i=1}^{E_n}\{f_{opt,i}\} \tag{32}$$

where E_n is the number of experiments and $f_{opt,i}$ represents the optimal value obtained in each individual run.

As shown in Fig. 5, the proposed L-WSONM algorithm surpasses the other eight algorithms across the majority of CEC2020 benchmark functions, securing the top overall rank. It achieves the lowest mean fitness values while maintaining consistently low standard deviations, reflecting stable and robust convergence behavior. Furthermore, L-WSONM demonstrates outstanding performance in both the exploration and exploitation phases of the search process.

Table 3 Analysis of Nelder–Mead trigger interval (α) effects on L-WSONM performance

Function	Index	L-WSONM ($\alpha = 10$)	L-WSONM($\alpha = 25$)	L-WSONM($\alpha = 50$)
F1	Mean	$2.0485e+03$	$2.2190e+03$	$1.5855e+03$
	Std	$2.1631e+03$	$2.3713e+03$	$2.2267e+03$
	Best	101.9668	107.5758	105.0400
	Worst	$7.8937e+03$	$1.1494e+04$	$9.8926e+03$
	Rank	2	3	1
F2	Mean	$1.8383e+03$	$1.8738e+03$	$1.9614e+03$
	Std	281.4539	310.6900	294.2206
	Best	$1.3250e+03$	$1.3605e+03$	$1.3610e+03$
	Worst	$2.4889e+03$	$2.6137e+03$	$2.5232e+03$
	Rank	1	2	3
F3	Mean	732.8942	735.42	738.19
	Std	11.7316	11.682	15.876
	Best	715.7485	717.16	717.94
	Worst	769.7866	760.68	774.59
	Rank	1	2	3
F4	Mean	$1.9011e+03$	$1.9015e+03$	$1.9018e+03$
	Std	0.5351	0.98479	1.0259
	Best	$1.9001e+03$	$1.9005e+03$	$1.9005e+03$
	Worst	$1.9020e+03$	$1.9041e+03$	$1.9052e+03$
	Rank	1	2	3
F5	Mean	$2.0139e+03$	$2.4268e+03$	$2.8198e+03$
	Std	148.8277	$1.1855e+03$	$1.7159e+03$
	$Best$	$1.7392e+03$	$1.7064e+03$	$1.8087e+03$
	$Worst$	$2.3869e+03$	$8.2868e+03$	$9.6828e+03$
	$Rank$	1	2	3
F6	Mean	$1.6033e+03$	$1.6069e+03$	$1.6056e+03$
	Std	6.4056	12.362	7.769
	Best	1600	1600	1600
	Worst	$1.6188e+03$	$1.659e+03$	$1.6188e+03$
	Rank	1	3	2
F7	Mean	$2.1962e+03$	$2.2559e+03$	$2.2692e+03$
	Std	93.4660	152.72	130.62
	Best	$2.1001e+03$	$2.1004e+03$	$2.1006e+03$
	Worst	$2.4674e+03$	$2.7445e+03$	$2.5376e+03$
	Rank	1	2	3
F8	Mean	$2.2998e+03$	$2.2997e+03$	$2.3004e+03$
	Std	13.1420	13.513	9.7273
	Best	$2.2310e+03$	$2.2286e+03$	$2.2492e+03$
	Worst	$2.3099e+03$	$2.307e+03$	$2.3043e+03$
	Rank	2	1	3
F9	Mean	$2.6882e+03$	$2.7127e+03$	$2.7052e+03$
	Std	116.5157	97.553	104.69
	Best	2500	2500	2500
	Worst	$2.8200e+03$	$2.7848e+03$	$2.7765e+03$
	Rank	1	3	2
F10	Mean	$2.9222e+03$	$2.9369e+03$	$2.9267e+03$
	Std	23.7697	17.454	23.355
	Best	$2.8977e+03$	$2.8977e+03$	$2.8977e+03$
	Worst	$2.9502e+03$	$2.9469e+03$	$2.9499e+03$
	Rank	1	3	2
Sum Rank		12	25	25
Mean Rank		1.2	2.5	2.5
Total Rank		1	2	3

Table 4 Ablation study evaluating the effects of the Nelder–Mead and Lévy flight mechanisms on the performance of L-WSONM

Function	Index	L-WSONM($\alpha = 10$)	WSO	WSONM ($\alpha = 10$)	L-WSO
F1	Mean	$2.0485e+03$	$3.9654e+06$	$1.3533e+06$	$1.9158e+03$
	Std	$2.1631e+03$	$1.0924e+07$	$5.1388e+06$	$2.8606e+03$
	Best	101.9668	391.0567	146.8	100.24
	Worst	$7.8937e+03$	$5.0060e+07$	$2.7061e+07$	$12.184e+03$
	Avg. Runtime (s)	0.3432	0.0717	0.2532	0.1230
	Rank	2	4	3	1
F2	Mean	$1.8383e+03$	$2.0644e+03$	$1.4743e+03$	$1.900e+03$
	Std	281.4539	444.7421	214.86	246.78
	Best	$1.3250e+03$	$1.1072e+03$	$1.1008e+03$	$1.364e+03$
	Worst	$2.4889e+03$	$2.7421e+03$	$1.9013e+03$	$2.4466e+03$
	Avg. Runtime (s)	0.4998	0.0518	0.3758	0.1274
	Rank	2	4	1	3
F3	Mean	732.8942	734.7150	722.53	738.94
	Std	11.7316	11.1971	6.6625	15.935
	Best	715.7485	716.3456	713.64	715.97
	Worst	769.7866	757.6640	738.64	801.26
	Avg. Runtime (s)	0.4661	0.0806	0.3424	0.1405
	Rank	2	3	1	4
F4	Mean	$1.9011e+03$	$1.9098e+03$	$1.9026e+03$	$1.9019e+03$
	Std	0.5351	26.0477	1.4103	1.2714
	Best	$1.9001e+03$	$1.9009e+03$	$1.9005e+03$	$1.9001e+03$
	Worst	$1.9020e+03$	$2.0446e+03$	$1.9058e+03$	$1.9052e+03$
	Avg. Runtime (s)	0.4317	0.0747	0.3241	0.1368
	Rank	1	4	3	2
F5	Mean	$2.0139e+03$	$2.1264e+03$	$1.9823e+03$	$6.046e+03$
	Std	148.8277	341.9680	161.62	$5.4157e+03$
	Best	$1.7392e+03$	$1.8437e+03$	$1.7501e+03$	$1.8205e+03$
	Worst	$2.3869e+03$	$3.6733e+03$	$2.4362e+03$	$25.538e+03$
	Avg. Runtime (s)	0.4368	0.0728	0.3311	0.1392
	Rank	2	3	1	4
F6	Mean	$1.6033e+03$	$1.6050e+03$	$1.602e+03$	$1.6081e+03$
	Std	6.4056	6.9428	5.0903	12.348
	Best	1600	$1.600e+03$	$1.600e+03$	$1.600e+03$
	Worst	$1.6188e+03$	$1.6171e+03$	$1.6171e+03$	$1.6585e+03$
	Avg. Runtime (s)	0.4115	0.0676	0.2976	0.1339
	Rank	2	3	1	4
F7	Mean	$2.1962e+03$	$2.2113e+03$	$2.1923e+03$	$10.755e+03$
	Std	93.4660	135.4759	100.14	$11.377e+03$
	Best	$2.1001e+03$	$2.1024e+03$	$2.1009e+03$	$2.2617e+03$
	Worst	$2.4674e+03$	$2.7640e+03$	$2.5086e+03$	$59.571e+03$
	Avg. Runtime (s)	0.4167	0.0718	0.3173	0.1385
	Rank	1	2	3	4
F8	Mean	$2.2998e+03$	$2.3082e+03$	$2.1923e+03$	$2.3006e+03$
	Std	13.1420	6.6634	100.14	11.623
	Best	$2.2310e+03$	$2.3012e+03$	$2.1009e+03$	$2.2395e+03$
	Worst	$2.3099e+03$	$2.3319e+03$	$2.5086e+03$	$2.3062e+03$
	Avg. Runtime (s)	0.8550	0.1037	0.6073	0.1900
	Rank	2	4	1	3
F9	Mean	$2.6882e+03$	$2.6244e+03$	$2.6943e+03$	$2.7256e+03$
	Std	116.5157	126.1103	102.43	90.835
	Best	2500	$2.5008e+03$	$2.5009e+03$	$2.500e+03$
	Worst	$2.8200e+03$	$2.7906e+03$	$2.8064e+03$	$2.7839e+03$
	Avg. Runtime (s)	0.8343	0.1073	0.6698	0.3184
	Rank	2	1	3	4

Table 4 continued

Function	Index	L-WSONM($\alpha = 10$)	WSO	WSONM ($\alpha = 10$)	L-WSO
F10	Mean	$2.9222e+03$	$2.9225e+03$	$2.916e+03$	$2.9332e+03$
	Std	23.7697	22.7915	17.694	21.019
	Best	$2.8977e+03$	$2.8978e+03$	$2.8978e+03$	$2.8977e+03$
	Worst	$2.9502e+03$	$2.9563e+03$	$2.9463e+03$	$2.9497e+03$
	Avg. Runtime (s)	0.7041	0.0959	0.5596	0.1814
	Rank	2	3	1	4
Sum Rank		18	31	18	33
Mean Rank		1.8	3.1	1.8	3.3
Total Rank		1	2	1	3

Table 6 reports the results of the nonparametric Wilcoxon signed-rank test, which assesses whether the proposed algorithm exhibits statistically significant differences in performance at the 5% significance level compared to its competitors. In this table, a symbol + denotes that L-WSONM significantly outperforms a specific algorithm, a − indicates significantly worse performance, and an = denotes no statistically significant difference.The *ranksum* command was used to evaluate our results. This nonparametric test does not assume normality of the data and is suitable for relatively small to moderate sample sizes (30 independent runs per function). Extremely small p-values indicate that the proposed method consistently outperforms the competitors, confirming statistical significance. Additionally, the table includes the Overall Effectiveness (OE) metric, which summarizes the relative performance of L-WSONM against all competitors. The OE metric is computed as follows

$$OE = (T_N - T_L)/T_N) \times 100 \tag{33}$$

Here, T_N is fixed at 10, while T_L denotes the number of unsuccessful evaluations for each algorithm. The proposed L-WSONM algorithm attains an Overall Effectiveness (OE) score of 100% when compared to POA, GJO, RSA, WSO, TSA, GA, PDO, and SOA, and achieves a 90% OE score against the GWO algorithm.

4 Cruise control system

The vehicle speed v is regulated to follow the reference speed v_{ref} by modulating the engine throttle angle u. The longitudinal dynamics of the vehicle are described by Eq. (34), where The terms F_d, F_a, and F_g refer to the driving force produced by the engine, the aerodynamic drag force, and the resistance due to slope, respectively.

$$F_d = M\frac{dv}{dt} + F_a + F_g \tag{34}$$

Inertial resistance is modeled by the expression $M\frac{dv}{dt}$. A schematic diagram of the vehicle cruise control system is displayed in Fig. 6, in which θ, v_w, C_a, and M represent the road angle, wind gust speed, aerodynamic drag coefficient, and total mass, respectively. This paper models the vehicle's state using Eqs. (35) and (36) neglecting wind gust speed and climbing resistance [52].

$$\dot{v} = \frac{1}{M}\left(F_d - C_a v^2\right) \tag{35}$$

$$\dot{F}_d = \frac{1}{T}\left(C_1 u(t-\tau) - F_d\right) \tag{36}$$

To maintain equilibrium at the nominal operating speed of $v_0 = 30$ km/h, a nominal driving force F_{d0} and a corresponding nominal throttle position u_0 are given as follows:

$$F_{d0} = C_a v_0^2 \tag{37}$$

$$u_0 = \frac{F_{d0}}{C_1} \tag{38}$$

The following expressions are derived by linearizing the nonlinear cruise control system model around the specified operating points:

$$\delta\dot{v} = -\frac{2C_a v_0}{M}\delta v + \frac{1}{M}\delta F_d \tag{39}$$

$$\delta\dot{F}_d = -\frac{1}{T}\delta F_d + \frac{C_1}{T}\delta u(t-\tau) \tag{40}$$

Based on the linearized dynamics, the transfer function of the vehicle cruise control system is obtained as follows:

$$\begin{aligned} G(s) = \frac{\Delta V(s)}{\Delta U(s)} &= \frac{C}{(s-p_1)(s-p_2)(s-p_3)} \\ &= \frac{2.4767}{(s+0.0476)(s+1)(s+5)} \end{aligned} \tag{41}$$

where $C = \frac{C_1}{MT}$, $p_1 = -\frac{2C_a v_0}{M}$, $p_2 = -\frac{1}{T}$, $p_3 = -\frac{1}{\tau}$, and the parameters of the cruise control system are listed in Table 7.

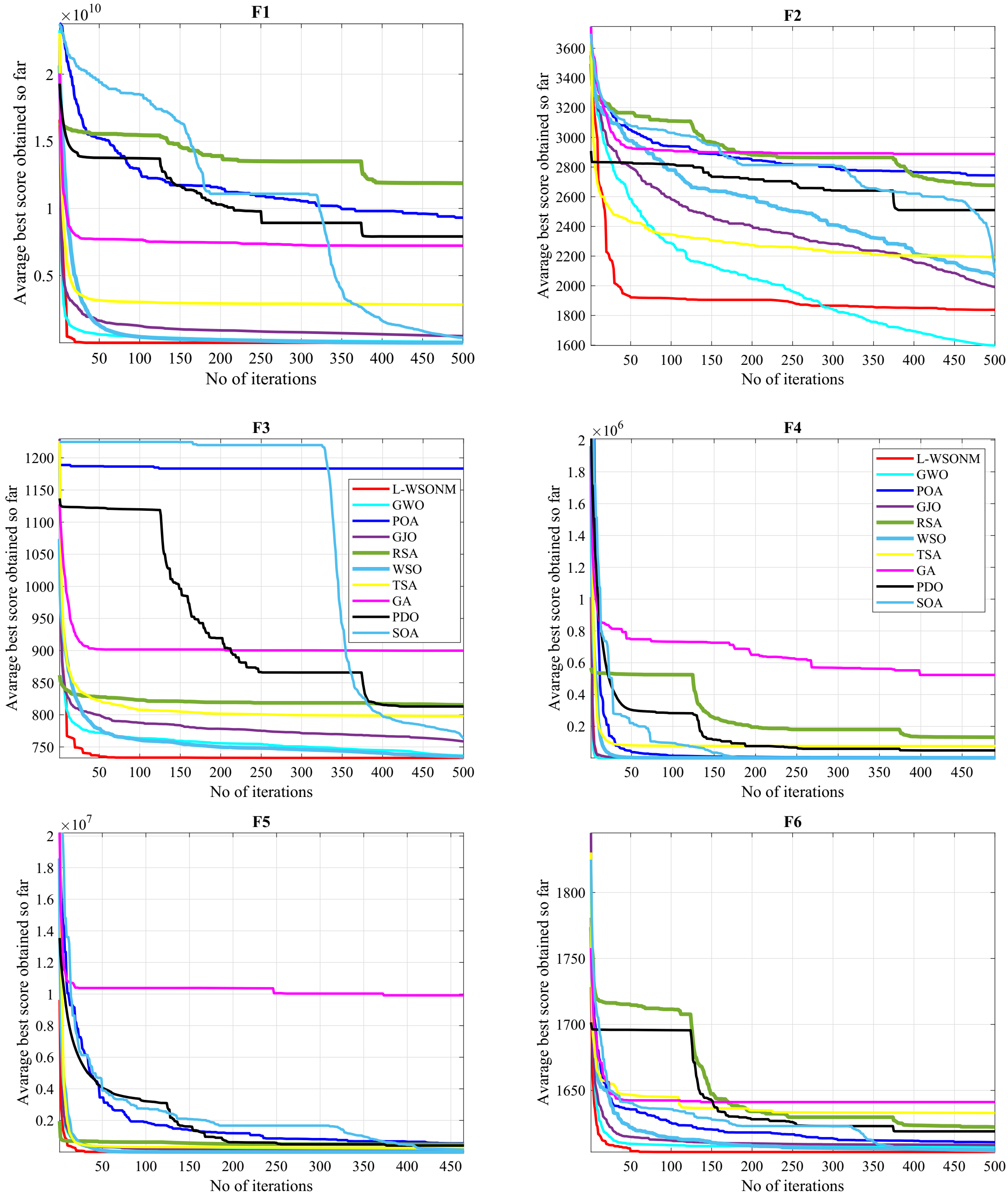

Fig. 4 Convergence behavior of the different algorithms on the test functions (F1–F10)

This transfer function accurately captures the dynamics of the linear approximation of the vehicle cruise control system, serving as a foundation for analyzing and designing control strategies aimed at precise speed regulation and enhanced driving performance.

5 L-WSONM algorithm-based RPIDD^2 controller design

In a standard PID controller, the output is shaped by three tuning constants: proportional (K_p), integral (K_i), and derivative (K_d) gains. These parameters regulate the influence of each component in the controller output and are mathematically

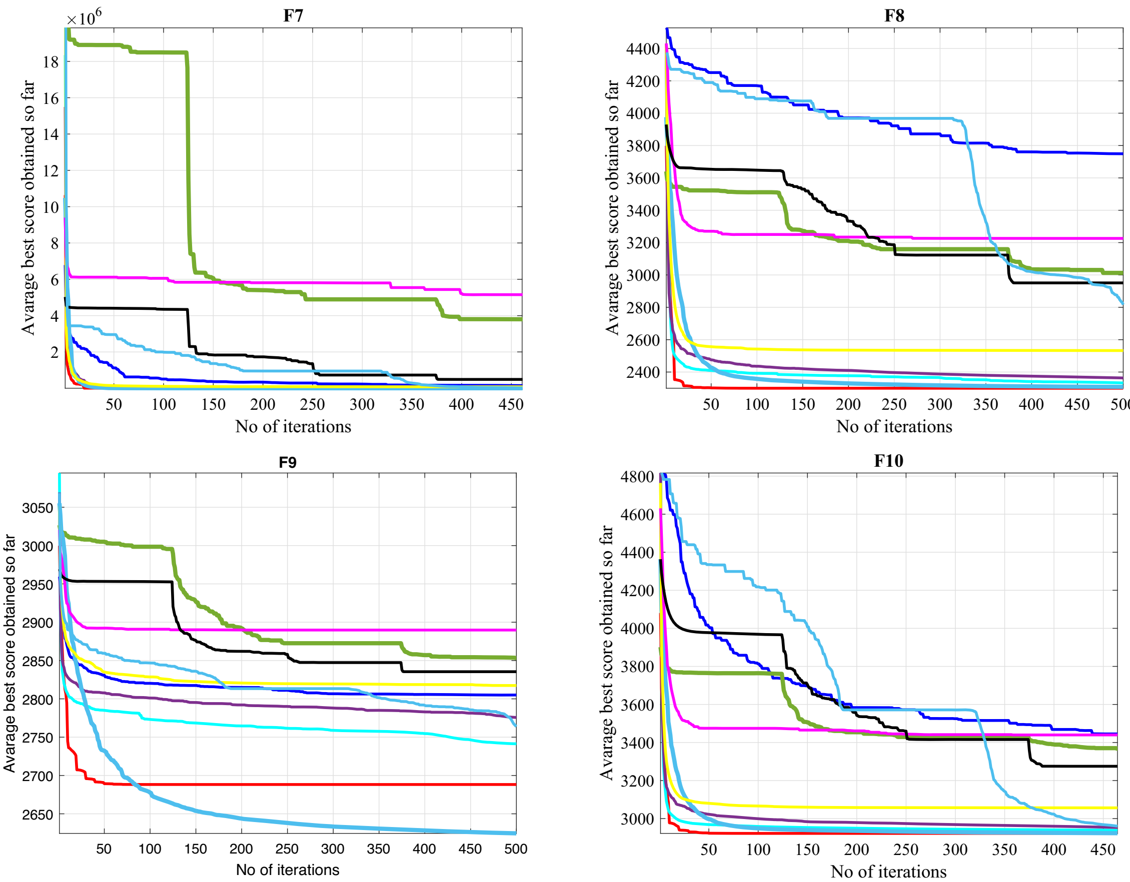

Fig. 4 continued

represented as follows[57]:

$$C_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \tag{42}$$

The conventional PID controller is widely applied in numerous domains. Nevertheless, for dynamic applications like cruise control systems, greater flexibility is often required. As a result, various modified forms of the PID controller have been investigated. An advanced form of the PID controller, known as the RPIDD^2 controller, has shown enhanced performance in terms of system stability and steady-state precision [3]. This controller is derived by augmenting the standard PID structure with a second-order derivative term. To mitigate the effects of high-frequency noise, low-pass filters are also integrated into the derivative components. These additions help reduce the steady-state error and enhance the overall stability of the system. The RPIDD^2 controller's transfer function is defined as:

$$C_{\mathrm{RPIDD}^2}(s) = K_p + \frac{K_i}{s} + K_d\left(\frac{N_1 s}{s+N_1}\right) + K_{dd}\left(\frac{N_2 s}{s+N_2}\right)^2 \tag{43}$$

where K_p, K_i, K_d, and K_{dd} denote the gains associated with proportional, integral, derivative, and second-order derivative terms respectively, while N_1 and N_2 represent the filter coefficients. Figure 7 presents the internal structure of the RPIDD^2 controller.

As illustrated in Fig. 8, the vehicle cruise control system operates in a closed-loop configuration with the RPIDD^2 controller. In this configuration, the controller processes the error between the reference and actual vehicle speeds to produce control signals that regulate the throttle input. This integration enhances the system's ability to maintain the desired speed, while improving stability, transient response, and steady-state accuracy.

5.1 Objective function

In this study, the controller design problem is formulated as a minimization task to determine the optimal controller parameters. To achieve an effective and practical solution, the system's mathematical model is defined as follows:

$$\vec{K} = [K_p,\ K_i,\ K_d,\ K_{dd},\ N_1,\ N_2]' \tag{44}$$

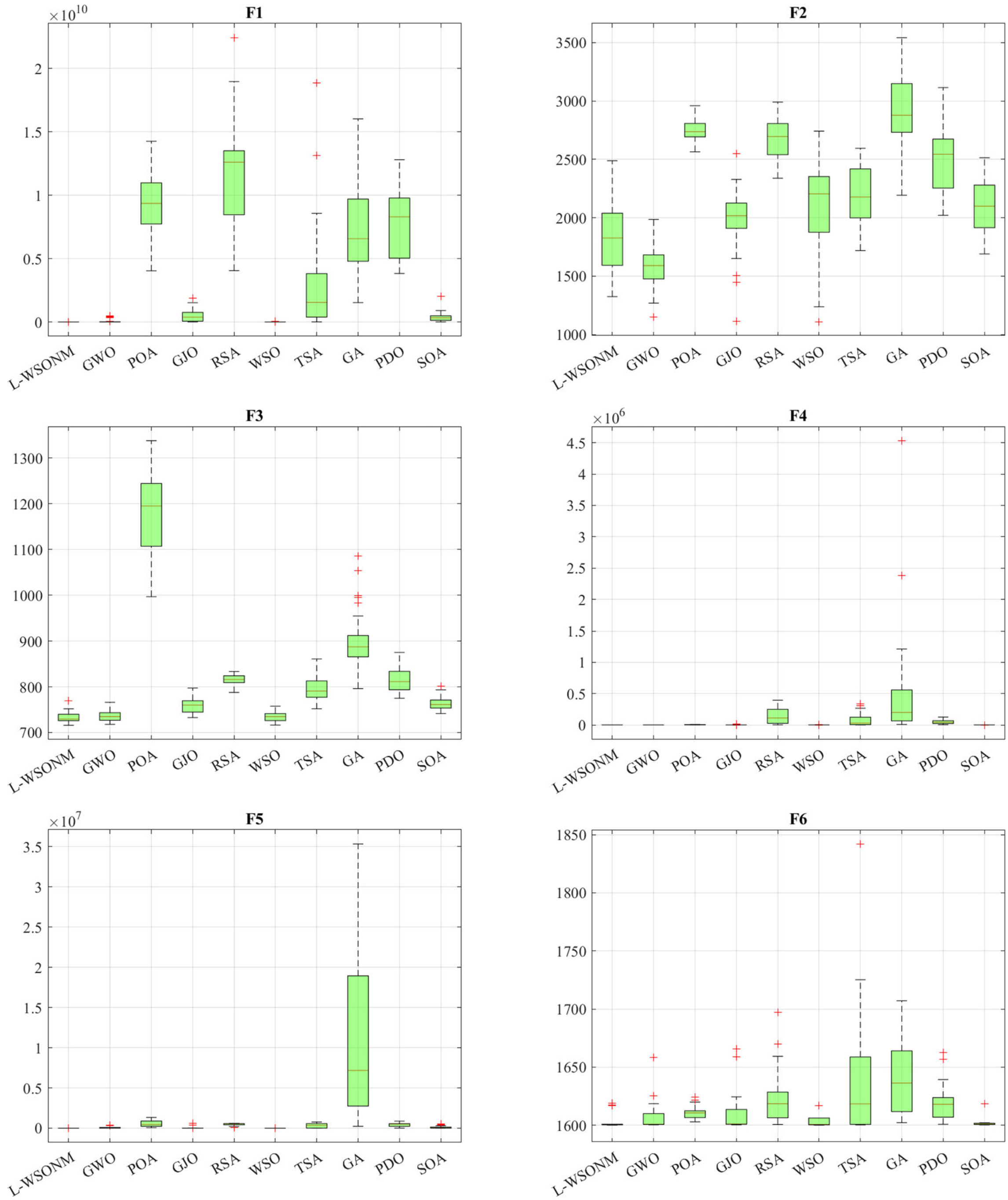

Fig. 5 Boxplot diagram of the proposed L-WSONM algorithm and alternative algorithms on the test functions (F1–F10)

In this study, the objective function $F(\vec{K})$ employed for controller performance evaluation is defined as follows [39]:

$$F(\vec{K}) = \left(1 - e^{-\rho}\right)\left(\frac{\%OS}{100} + E_{ss}\right) + e^{-\rho}(T_s - T_r) \quad (45)$$

This function provides a trade-off between maximum overshoot percentage (%OS), steady-state error (E_{ss}), settling time (T_s), and rise time (T_r), with ρ acting as a weighting coefficient. Based on simulation results, the optimal performance is achieved when the weighting coefficient is set to $\rho = 1$. Table 8 presents the selected parameters for the L-WSONM algorithm along with the defined boundaries for tuning the real PIDD^2 controller.

The tuning process of the real PIDD^2 controller using the proposed L-WSONM algorithm for the vehicle cruise control system is depicted in Fig. 9. According to the figure, the

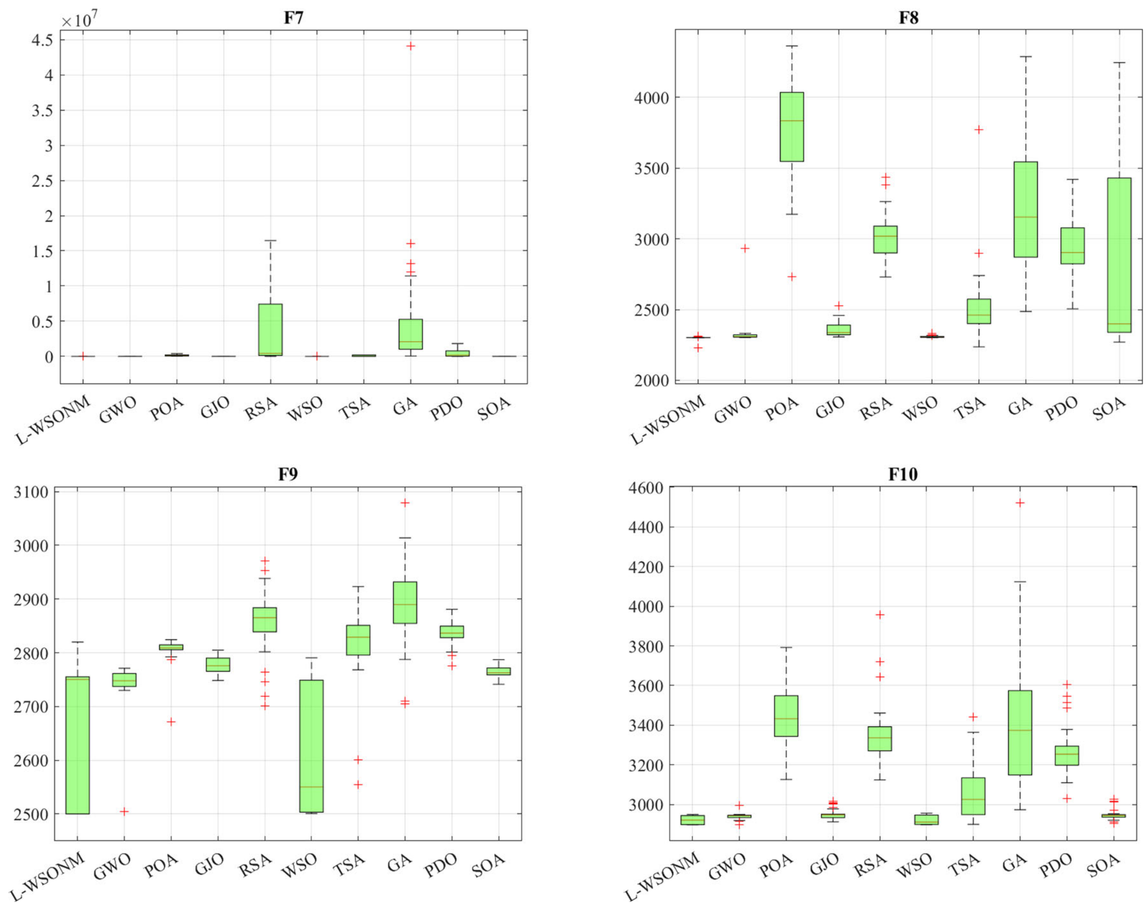

Fig. 5 continued

parameters of the L-WSONM algorithm are first initialized, followed by the generation of a random population of agents. The proposed algorithm then iteratively updates the parameters of the RPIDD^2 controller by optimizing the objective function, $F(\vec{K})$. The process iterates until the predefined maximum number of iterations is completed, ultimately producing the controller's optimal parameters.

5.2 Statistical assessment

An analysis was conducted to evaluate the efficiency of the proposed L-WSONM ($\alpha = 25$) method in comparison with the L-WSO, WSONM ($\alpha = 25$) and standard WSO algorithms. Each algorithm was independently executed 30 times using a population of 30 individuals and a maximum of 50 iterations. The resulting ZGL objective function values were used to calculate statistical indicators such as the mean, standard deviation, best, and worst outcomes. As depicted in Fig. 10, L-WSONM demonstrates consistently better performance, achieving lower ZGL values across all runs.

Table 9 presents the statistical performance comparison of the four algorithms, including mean, standard deviation, best, and worst values, along with the results of the Wilcoxon signed-rank test. These descriptive statistics, combined with the nonparametric test outcomes, provide a comprehensive assessment of algorithmic performance. Across all metrics, the proposed L-WSONM algorithm consistently outperforms the competing methods (WSO, WSONM, and L-WSO). In particular, L-WSONM achieves the lowest mean and best (minimum) values, indicating superior optimization capability. Its notably small standard deviation demonstrates greater stability and consistency across the 30 independent runs. Furthermore, the worst-case value of L-WSONM is also lower than those of the other algorithms, confirming its robustness even under challenging conditions. The results of the Wilcoxon signed-rank test further validate these findings: L-WSONM is marked as the statistical winner, whereas the competing algorithms yield significantly higher p-values. This provides strong evidence that the performance improvements obtained by L-WSONM are statistically significant.

The comparative performance of the WSO, WSONM, L-WSO, and L-WSONM algorithms is illustrated using boxplots in Fig. 11. These box-and-whisker plots visually summarize the distribution and variability of the metric values

Table 5 Results of F1–F10 benchmark functions

Function	Index	L-WSONM(Proposed)	GWO	POA	GJO	RSA	WSO	TSA	GA	PDO	SOA
	Mean	2.0485e + 03	7.1723e + 07	9.3042e + 09	4.9766e + 08	1.1882e + 10	3.9654e + 06	2.8398e + 09	7.2304e + 09	7.9144e + 09	3.7540e + 08
	Std	2.1631e + 03	1.5425e + 08	2.2867e + 09	4.7375e + 08	4.0444e + 09	1.0924e + 07	4.2071e + 09	3.7070e + 09	2.8295e + 09	3.9799e + 08
F1	Best	101.9668	2.0252e + 04	4.0309e + 09	6.1575e + 05	4.0534e + 09	391.0567	8.9479e + 06	1.5237e + 09	3.8243e + 09	4.3632e + 06
	Worst	7.8937e + 03	4.9755e + 08	1.4254e + 10	1.8733e + 09	2.2420e + 10	5.0060e + 07	1.8862e + 10	1.6016e + 10	1.2803e + 10	2.0454e + 09
	Rank	1	3	9	5	10	2	6	7	8	4
	Mean	1.8383e + 03	1.5976e + 03	2.7439e + 03	1.9915e + 03	2.6773e + 03	2.0644e + 03	2.1930e + 03	2.8882e + 03	2.5098e + 03	2.1008e + 03
	Std	281.4539	215.4630	99.5912	289.7017	184.1615	444.7421	264.7313	341.1535	298.3290	231.3663
F2	Best	1.3250e + 03	1.1502e + 03	2.5650e + 03	1.1134e + 03	2.3387e + 03	1.1072e + 03	1.7193e + 03	2.1927e + 03	2.0214e + 03	11.6904e + 03
	Worst	2.4889e + 03	1.9855e + 03	2.9599e + 03	2.5506e + 03	2.9908e + 03	2.7421e + 03	2.5951e + 03	3.5423e + 03	3.1150e + 03	2.5142e + 03
	Rank	2	1	9	3	8	4	6	10	7	5
	Mean	732.8942	736.4794	1.1834e + 03	758.7837	815.5392	734.7150	798.0104	899.9011	812.9227	763.3988
	Std	11.7316	12.8255	91.3953	16.3581	11.3154	11.1971	27.2056	66.8779	22.9749	14.7811
F3	Best	715.7485	717.8110	996.6813	732.6740	787.6450	716.3456	751.8153	796.1478	775.0100	741.5333
	Worst	769.7866	765.9928	1.3380e + 03	797.2236	833.6038	757.6640	860.9451	1.0854e + 03	875.2954	801.3499
	Rank	1	3	10	4	8	2	6	9	7	5
	Mean	1.9011e + 03	1.9028e + 03	5.2572e + 03	3.1564e + 03	1.3187e + 05	1.9098e + 03	7.6402e + 04	5.2343e + 05	4.9036e + 04	1.9060e + 03
	Std	0.5351	0.9630	1.9831e + 03	3.7309e + 03	1.1768e + 05	26.0477	1.0046e + 05	9.0596e + 05	3.4039e + 04	3.8298
F4	Best	1.9001e + 03	1.9010e + 03	2.3552e + 03	1.9019e + 03	3.0813e + 03	1.9009e + 03	1.9042e + 03	9.4067e + 03	4.1842e + 03	1.9027e + 03
	Worst	1.9020e + 03	1.9045e + 03	9.5704e + 03	1.6954e + 04	3.9634e + 05	2.0446e + 03	3.3449e + 05	4.5321e + 06	1.2672e + 05	1.9235e + 03
	Rank	1	2	6	5	9	4	8	10	7	3
	Mean	2.0139e + 03	9.3633e + 04	5.3755e + 05	3.9112e + 04	4.8090e + 05	2.1264e + 03	2.6470e + 05	9.9161e + 06	4.0771e + 05	1.0917e + 05
	Std	148.8277	1.5046e + 05	3.8268e + 05	1.2543e + 05	1.2657e + 05	341.9680	2.6169e + 05	9.4724e + 06	2.2486e + 05	1.6111e + 05
F5	Best	1.7392e + 03	3.2554e + 03	5.6975e + 04	3.7859e + 03	5.7689e + 04	1.8437e + 03	4.4435e + 03	2.4134e + 05	1.4418e + 04	4.2902e + 03
	Worst	2.3869e + 03	4.0868e + 05	1.3309e + 06	6.1343e + 05	6.0870e + 05	3.6733e + 03	7.6286e + 05	3.5345e + 07	8.7267e + 05	5.1388e + 05
	Rank	1	4	9	3	8	2	6	10	7	5
	Mean	1.6033e + 03	1.6069e + 03	1.6107e + 03	1.6086e + 03	1.6223e + 03	1.6050e + 03	1.6329e + 03	1.6412e + 03	1.6190e + 03	1.6038e + 03
	Std	6.4056	12.1708	5.5663	16.2793	22.0800	6.9428	49.2334	29.2187	15.6925	6.6469
F6	Best	1600	1.6003e + 03	1.6029e + 03	1.6003e + 03	1.6006e + 03	1.6000e + 03	1.6003e + 03	1.6022e + 03	1.6008e + 03	1.6003e + 03
	Worst	1.6188e + 03	1.6585e + 03	1.6242e + 03	1.6659e + 03	1.6973e + 03	1.6171e + 03	1.8420e + 03	1.7070e + 03	1.6623e + 03	1.6186e + 03
	Rank	1	4	6	5	8	3	9	10	7	2
	Mean	2.1962e + 03	9.0457e + 03	1.4259e + 05	9.4795e + 03	3.8009e + 06	2.2113e + 03	8.2188e + 04	5.1191e + 06	4.8824e + 05	1.2389e + 04
	Std	93.4660	5.9898e + 03	9.6707e + 04	4.9878e + 03	5.5123e + 06	135.4759	9.5243e + 04	8.5197e + 06	6.0097e + 05	7.4933e + 03
F7	Best	2.1001e + 03	2.7128e + 03	2.3411e + 04	3.1466e + 03	8.9028e + 03	2.1024e + 03	2.9861e + 03	6.4174e + 04	7.2919e + 03	4.5420e + 03
	Worst	2.4674e + 03	2.5786e + 04	4.0065e + 05	2.0306e + 04	1.6462e + 07	2.7640e + 03	2.0827e + 05	4.4105e + 07	1.8327e + 06	2.8425e + 04
	Rank	1	3	7	4	9	2	6	10	8	5

Table 5 continued

Function	Index	L-WSONM(Proposed)	GWO	POA	GJO	RSA	WSO	TSA	GA	PDO	SOA
	Mean	2.2998e + 03	2.3323e + 03	3.7493e + 03	2.3628e + 03	3.0128e + 03	2.3082e + 03	2.5327e + 03	3.2260e + 03	2.9510e + 03	2.8164e + 03
	Std	13.1420	113.7084	377.2345	54.1627	173.3208	6.6634	271.6377	472.3602	218.4052	687.8801
F8	Best	2.2310e + 03	2.3018e + 03	2.7313e + 03	2.3064e + 03	2.7304e + 03	2.3012e + 03	2.2366e + 03	2.4865e + 03	2.5051e + 03	2.2701e + 03
	Worst	2.3099e + 03	2.9325e + 03	4.3638e + 03	2.5247e + 03	3.4332e + 03	2.3319e + 03	3.7729e + 03	4.2882e + 03	3.4201e + 03	4.2456e + 03
	Rank	1	3	10	4	8	2	5	9	7	6
	Mean	2.6882e + 03	2.7414e + 03	2.8050e + 03	2.7757e + 03	2.8533e + 03	2.6244e + 03	2.8174e + 03	2.8898e + 03	2.8354e + 03	2.7639e + 03
	Std	116.5157	46.4996	26.5250	16.5063	61.0116	126.1103	76.3844	80.1375	22.2676	10.9687
F9	Best	2500	2.5037e + 03	2.6719e + 03	2.7483e + 03	2.7009e + 03	2.5008e + 03	2.5542e + 03	2.7044e + 03	2.7751e + 03	2.7414e + 03
	Worst	2.8200e + 03	2.7712e + 03	2.8244e + 03	2.8049e + 03	2.9707e + 03	2.7906e + 03	2.9232e + 03	3.0793e + 03	2.8809e + 03	2.7871e + 03
	Rank	2	3	6	5	9	1	7	10	8	4
	Mean	2.9222e + 03	2.9386e + 03	3.4452e + 03	2.9510e + 03	3.3649e + 03	2.9225e + 03	3.0564e + 03	3.4397e + 03	3.2750e + 03	2.9489e + 03
	Std	23.7697	17.4399	158.3523	28.9322	163.0172	22.7915	137.9578	366.5283	128.0468	27.8834
F10	Best	2.8977e + 03	2.8984e + 03	3.1263e + 03	2.9131e + 03	3.1242e + 03	2.8978e + 03	2.9003e + 03	2.9739e + 03	3.0294e + 03	2.9050e + 03
	Worst	2.9502e + 03	2.9971e + 03	3.7919e + 03	3.0154e + 03	3.9567e + 03	2.9563e + 03	3.4401e + 03	4.5213e + 03	3.6054e + 03	3.0281e + 03
	Rank	1	3	10	5	8	2	6	9	7	4
Sum Rank		12	29	82	43	85	22	65	94	73	43
Mean Rank		1.2	2.9	8.2	4.3	8.5	2.2	6.5	9.4	7.3	4.3
Total Rank		1	3	7	4	8	2	5	9	6	4

Table 6 Wilcoxon rank-sum test results of F1–F10 benchmark functions

L-WSONM(Proposed)	GWO		POA		GJO		RSA		WSO	
Function	*p*-value	Win	*p*-value	Win	*p*-value	Win	*p*-value	Win	*p*-value	Win
F1	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	2.0152e−08	+
F2	0.0015	−	3.0199e−11	+	0.0199	+	8.9934e−11	+	0.0117	+
F3	0.2340	=	3.0199e−11	+	9.0632e−08	+	3.0199e−11	+	0.4643	=
F4	2.6015e−08	+	3.0199e−11	+	4.0772e−11	+	3.0199e−11	+	1.1023e−08	+
F5	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	0.2062	=
F6	0.0022	+	1.6754e−06	+	7.5742e−04	+	6.2725e−08	+	0.4819	=
F7	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	0.5895	=
F8	1.8500e−08	+	3.0199e−11	+	3.3384e−11	+	3.0199e−11	+	1.8731e−07	+
F9	0.4822	=	2.8403e−09	+	1.0866e−06	+	1.3930e−08	+	0.2577	=
F10	0.0117	+	3.0161e−11	+	2.5296e−04	+	3.0161e−11	+	0.2772	=
Total ±/=	7/1/2		10/0/0		10/0/0		10/0/0		4/0/6	
OE	%90		%100		%100		%100		%100	

FGWO-WaOA-NM(Proposed)	TSA		GA		PDO		SOA	
Function	*p*-value	Win	*p*-value	Win	*p*-value	Win	*p*-value	Win
F1	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+
F2	3.3681e−05	+	7.3891e−11	+	2.9215e−09	+	3.7704e−04	+
F3	4.0772e−11	+	3.0199e−11	+	3.0199e−11	+	1.5465e−09	+
F4	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+
F5	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+
F6	5.8376e−05	+	2.1216e−09	+	2.7149e−08	+	0.0094	+
F7	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+	3.0199e−11	+
F8	5.0723e−10	+	3.0199e−11	+	3.0199e−11	+	5.0723e−10	+
F9	7.5677e−09	+	2.5949e−09	+	6.4721e−11	+	1.3161e−05	+
F10	8.3451e−08	+	3.0161e−11	+	3.0161e−11	+	0.0022	+
Total ±/=	10/0/0		10/0/0		10/0/0		10/0/0	
OE	%100		%100		%100		%100	

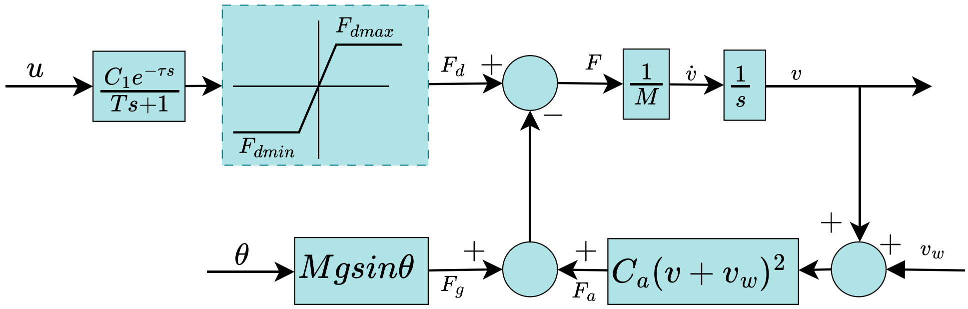

Fig. 6 Vehicle cruise control system dynamics [3]

Table 7 The parameter values of the vehicle cruise control system (adapted from [22])

Parameter	Definition	Value	Unit
C_a	Aerodynamic drag coefficient	1.19	N/(m/s)^2
C_1	Actuator constant	743	–
T	Observation time	1	sec
τ	Driver reaction time	0.2	sec
M	Mass of vehicle and passenger	1500	kg
g	Gravitational acceleration	9.8	m/s^2
$F_{d,\min}$	Minimum saturation limit	– 3500	N
$F_{d,\max}$	Maximum saturation limit	3500	N

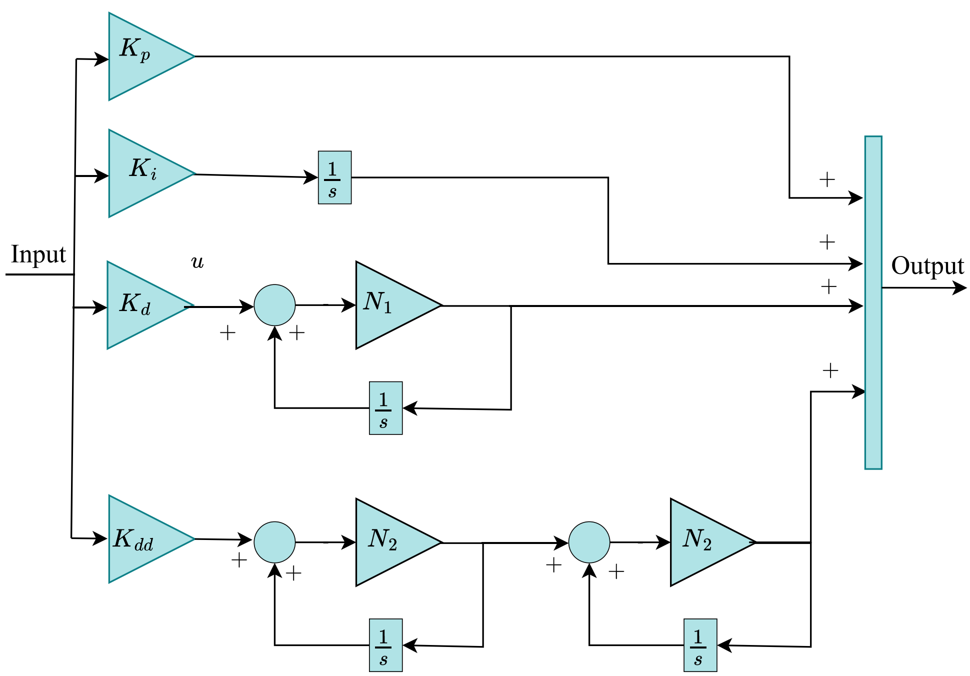

Fig. 7 Block diagram of real RPIDD^2 controller

Fig. 8 RPIDD^2 controlled vehicle cruise control system

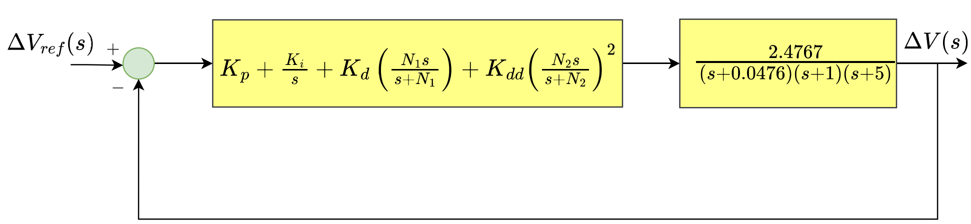

Table 8 Design boundaries of the RPIDD^2 controller and parameter settings of the L-WSONM algorithm

Parameters	Value
Lower bounds for $[K_p\ K_i\ K_d\ K_{dd}\ N_1\ N_2]$	[1, 0.1,1,0.1,100,100]
Upper bounds for $[K_p\ K_i\ K_d\ K_{dd}\ N_1\ N_2]$	[10, 1,10,1,1000,1000]
Population size	30
Total number of iterations	50
Variable size	6
Constant, a_0	6.25
Constant, a_1	100
Constant, a_2	0.0005
Maximum frequency of the wavy motion, f_{max}	0.75
Minimum frequency of the wavy motion, f_{min}	0.07
Initial velocity, p_{max}	1.5
Subordinate velocity, p_{min}	0.5

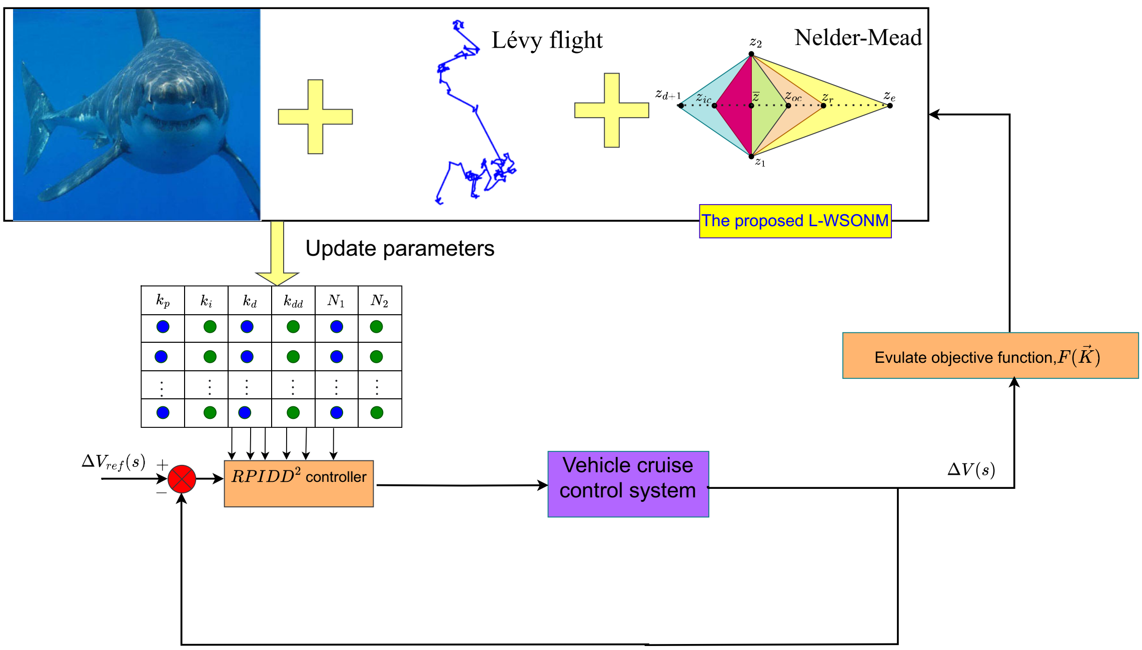

Fig. 9 Optimization of the controller's parameters for the vehicle cruise control system using the proposed L-WSONM

obtained from each algorithm. Notably, the L-WSONM boxplot exhibits a lower median and a more compact interquartile range compared with those of WSO, WSONM, and L-WSO. This indicates that L-WSONM not only delivers superior median performance but also demonstrates greater consistency across the 30 runs.

5.3 Convergence rate analysis

The convergence curve of the proposed method was evaluated in comparison to the original WSO-based approach. As illustrated in Fig. 12, the proposed L-WSONM algorithm is capable of reaching lower objective function values in the later stages of iteration and ultimately achieves a better minimum compared to the WSO optimizer. This demonstrates that, for the vehicle cruise control system, the proposed approach achieves more effective convergence than the original WSO algorithm (Figs. 13, 14).

6 Simulation results and discussion

For the analysis, controller parameters were extracted using WSO and L-WSONM. The best parameter set from WSO is: $K_p = 6.7891$, $K_i = 0.9815$, $K_d = 9.4496$, $K_{dd} = 0.8711$, $N_1 = 826.1547$, $N_2 = 281.0639$. These values lead to the closed-loop transfer function expressed in Eq. (46).

Table 9 Statistical measures and Wilcoxon signed-rank test analysis

Algorithm	Mean	Std	Best	Worst	Avg. Runtime (s)	Wilcoxon Test
WSO	0.0914	0.0037	0.0870	0.1010	30.27	p = 3.0199e−11
WSONM	0.0856	0.0027	0.0829	0.0937	78.36	p = 3.6459e−08
L-WSO	0.0865	0.0051	0.0828	0.1035	54.24	p = 2.8314e−08
L-WSONM (Proposed)	0.0830	3.41e−04	0.0827	0.0838	117.11	Winner

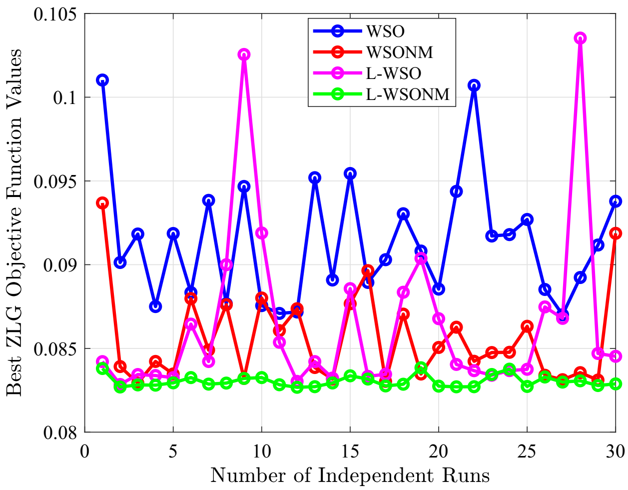

Fig. 10 Objective function ZGL values for each run

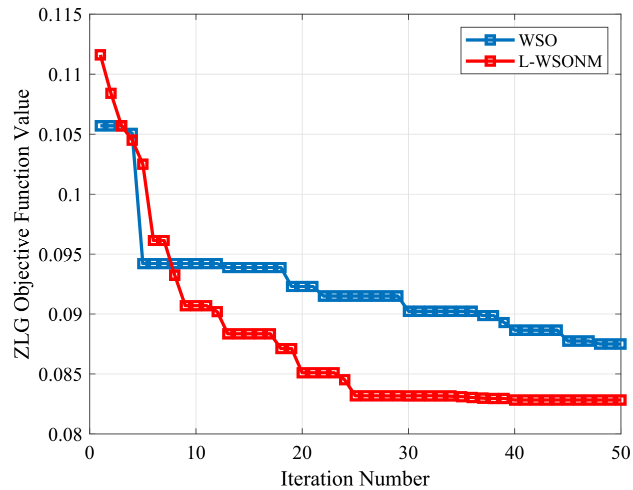

Fig. 12 Best convergence curves of WSO and L-WSONM

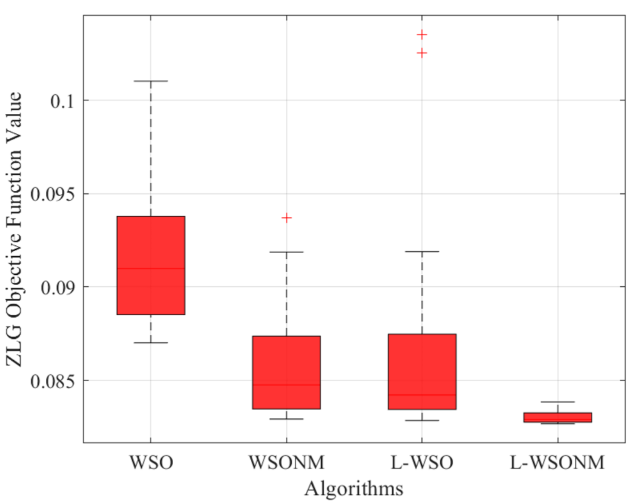

Fig. 11 Boxplot analysis of WSO and L-WSONM algorithms

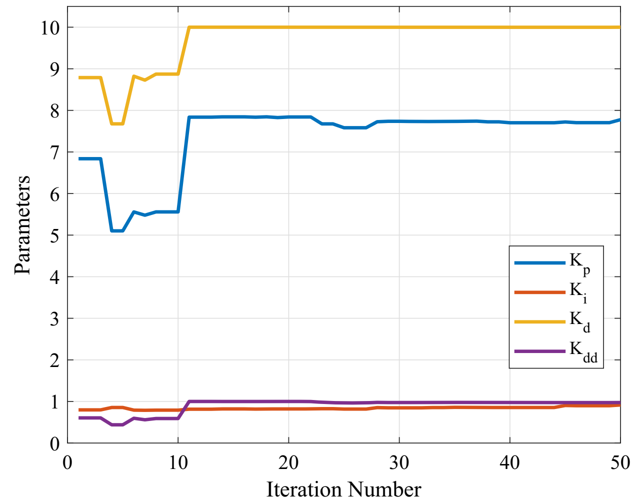

Fig. 13 Variation of K_p,K_i,K_d and K_{dd} parameters over iterations using the proposed L-WSONM algorithm

$$TF_{WSO} = \frac{1.898e05s^4 + 1.517e08s^3 + 1.537e09s^2 + 1.099e09s + 1.586e08}{s^7 + 1394s^6 + 5.518e05s^5 + 6.875e07s^4 + 5.493e08s^3 + 1.882e09s^2 + 1.114e09s + 1.586e08} \quad (46)$$

Likewise, the optimal controller parameters determined using the L-WSONM algorithm are: $K_p = 7.7760$, $K_i = 0.9153$, $K_d = 10.0000$, $K_{dd} = 0.9743$, $N_1 = 735.5033$, and $N_2 = 984.7011$ (see Figs. 13 and 14). Substituting these values into the transfer functions within the feedback con-

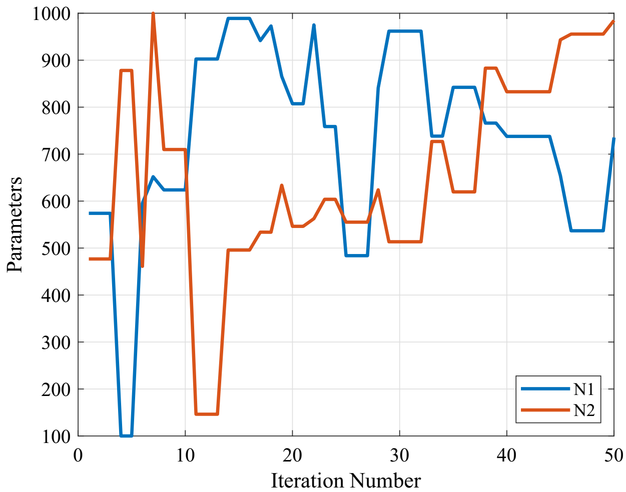

Fig. 14 Variation of $N1$ and $N2$ parameters over iterations using the proposed L-WSONM algorithm

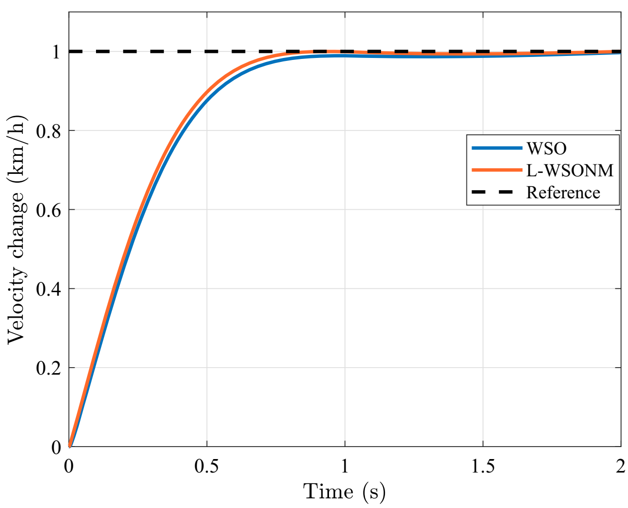

Fig. 15 Comparative step responses of velocity changes for the WSO- and L-WSONM-based real PIDD^2-controlled cruise control systems.

Table 10 Comparison of transient response specifications

Algorithm	OS (%)	t_r (s)	t_s (s)	Ess
WSO	0	0.488	0.784	0.0027474
L-WSONM (proposed)	0	0.463	0.692	0.00017022

figuration illustrated in Fig. 8 results in the corresponding closed-loop transfer function as given in Eq. (47). Figure 15 shows the comparative step responses of velocity changes for the WSO- and L-WSONM-based real PIDD^2-controlled cruise control systems.

Table 11 Comparison of frequency response specifications

Algorithm	WSO	L-WSONM (proposed)
Peak gain (dB)	1.0231	1.0157
Gain margin (dB)	47.7	58.1
Phase margin (deg)	171	172
Delay margin (s)	6.1235	5.9918
Bandwidth (Hz)	4.5425	4.7092

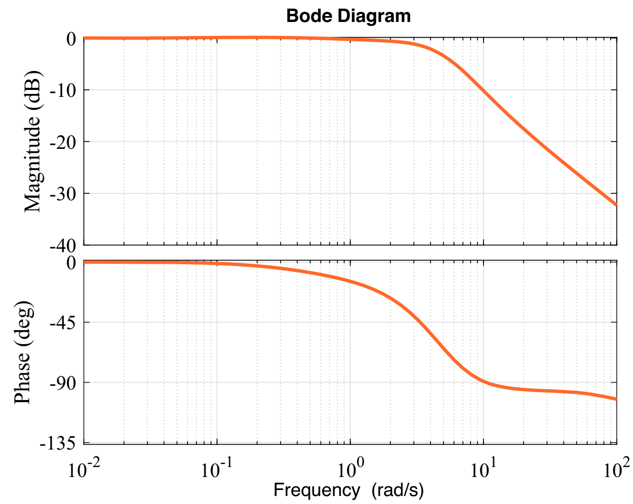

Fig. 16 The frequency response of the L-WSONM-based RPIDD^2-controlled vehicle cruise control system

Table 10 presents a numerical comparison of overshoot, rise time, settling time, and steady-state error for the vehicle cruise control system using the proposed L-WSONM algorithm-based RPIDD^2 controller versus the WSO algorithm-based controller. The developed controller demonstrates superior performance by significantly reducing the rise time, settling duration, and steady-state deviation.

The L-WSONM-based RPIDD^2 controller is additionally examined through frequency-domain evaluation. Accordingly, key frequency domain performance metrics including peak gain, gain margin, phase margin, delay margin, and bandwidth are reported in Table 11. Figure 16 presents the Bode plot of the vehicle cruise control system configured with the proposed controller.

6.1 Robustness test

To verify the robustness of the L-WSONM-based design, the system was extensively analyzed under diverse parameter

$$TF_{L-WSONM} = \frac{2.358e06s^4 + 1.757e09s^3 + 1.771e10s^2 + 1.374e10s + 1.617e09}{s^7 + 2711s^6 + 2.435e06s^5 + 7.302e08s^4 + 6.083e09s^3 + 2.148e10s^2 + 1.391e10s + 1.617e09} \quad (47)$$

Table 12 Performance analysis in the time domain under varying operational settings

Scenarios	Metrics	WSO	L-WSONM
Scenario 1 ($M = 1600$, $\tau = 0.18$)	t_r (s)	0.53	0.499
	t_s (s)	1.144	0.828
	OS (%)	0	0
	E_{ss}	0.003198	9.1437e−05
Scenario 2 ($M = 1550$, $\tau = 0.19$)	t_r (s)	0.507	0.479
	t_s (s)	0.899	0.75
	OS (%)	0	0
	E_{ss}	0.002928	9.5304e−05
Scenario 3 ($M = 1400$, $\tau = 0.22$)	t_r (s)	0.453	0.433
	t_s (s)	0.666	0.614
	OS (%)	0.34062	1.4618
	E_{ss}	0.0026148	0.00048865
Scenario 4 ($M = 1350$, $\tau = 0.16$)	t_r (s)	0.452	0.429
	t_s (s)	1.16	0.755
	OS (%)	0	0
	E_{ss}	0.0033787	0.0010315
Scenario 5 ($C_a = 0.9520$)	t_r (s)	0.485	0.461
	t_s (s)	0.767	0.684
	OS (%)	0	0
	E_{ss}	0.00034153	.0020263
Scenario 6 ($C_a = 1.4280$)	t_r (s)	00.49	0.465
	t_s (s)	0.804	0.702
	OS (%)	0	0
	E_{ss}	0.005142	0.0023572
Scenario 7 ($v_0 = 20$ km/h)	t_r (s)	0.483	0.459
	t_s (s)	0.757	0.678
	OS (%)	0.1269	0.011557
	E_{ss}	0.001269	0.0034962
Scenario 8 ($v_0 = 35$ km/h)	t_r (s)	0.49	0.465
	t_s (s)	0.8	0.7
	OS (%)	0	0
	E_{ss}	0.0047437	0.0019934

conditions. The time-domain performance metrics of the L-WSONM algorithm were compared with those of the original WSO algorithm. The analysis results offer important insights into the L-WSONM approach's ability to maintain favorable performance despite variations in parameters. Four unique scenarios were analyzed, each incorporating variations in the M and τ parameters to represent different operating conditions. Furthermore, two scenarios investigate a ±20% change in the aerodynamic drag coefficient, while the remaining two scenarios examine different velocity step responses of 20 km/h and 35 km/h. The time-domain performance metrics are presented in Table 12. Time-domain metrics demonstrate the consistent performance and robustness of the proposed L-WSONM-based controller under the presence of parameter variation. The largest deviations observed are 1.4618% for overshoot, 0.0340 s for rise time and 0.1360s for settling time. As the deviations stay within reasonable bounds, our L-WSONM-based controller demonstrates strong robustness to parameter changes. The L-WSONM algorithm demonstrated reliable and efficient system performance under varying parameters, emphasizing its robustness and stability.

Table 13 summarizes the gain margins (GM), phase margins (PM), and gain crossover frequencies (ω_{gc}) of the proposed RPIDD^2 controller across varying vehicle speeds and masses. The results indicate that the controller maintains high gain and phase margins for all considered operating points, suggesting robust stability under different vehicle dynamics.

The Status column, indicated by a checkmark (✓), confirms that the controller satisfies standard robustness criteria

Table 13 Gain and phase margins of the proposed RPIDD^2 controller across varying vehicle speeds and masses

Speed (km/h)	Mass (kg)	GM (dB)	PM (deg)	ω_{gc} (rad/s)	Status
30	1600	57.8	171	0.503	✓
30	1400	57.5	173	0.497	✓
30	1300	56.8	173	0.49	✓
40	1500	58.1	173	0.441	✓
20	1500	58.1	171	0.561	✓
15	1500	58.1	171	0.591	✓
40	1600	58.7	172	0.499	✓
20	1400	57.5	172	0.563	✓
15	1300	56.8	172	0.602	✓

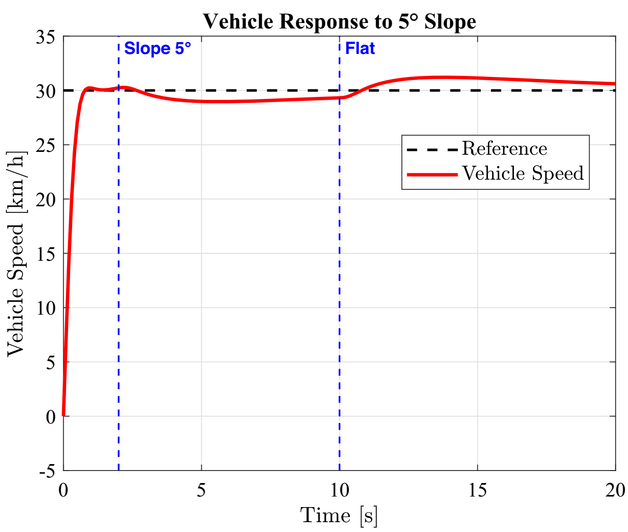

Fig. 17 Vehicle speed response under multiple slope disturbances

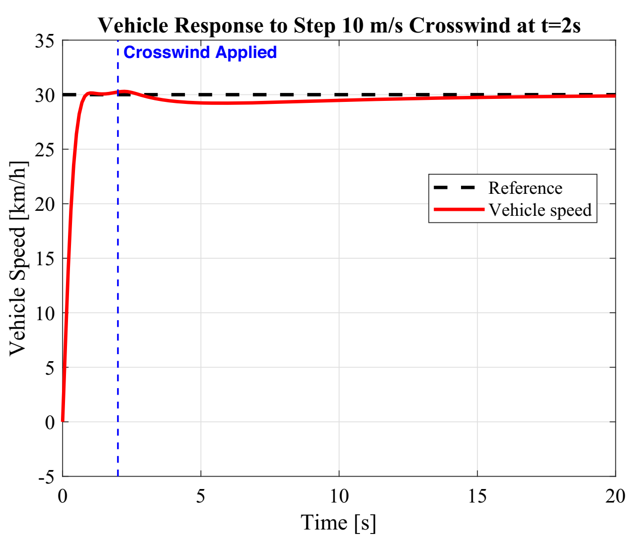

Fig. 18 Vehicle speed response under a 10 m/s crosswind disturbance

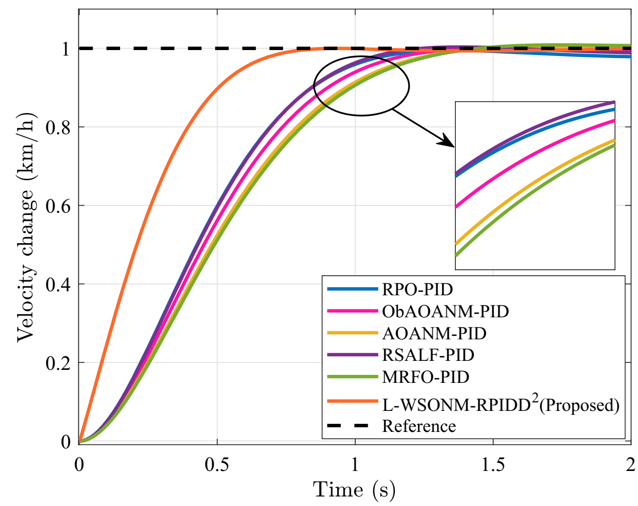

Fig. 19 Comparison of the closed-loop system's step responses with different optimized PID controllers and the proposed approach

(e.g., GM $>$ 6 dB, PM $>$ 30°) across the tested range of speeds (15–40 km/h) and vehicle masses (1300–1600 kg). These results demonstrate that the RPIDD^2 controller consistently maintains robust stability and adequate performance despite variations in vehicle operating conditions.

6.2 Sensitivity analysis to road slope

The L-WSONM-based RPIDD^2-controlled vehicle cruise system is further evaluated for slope disturbance rejection. Figure 17 illustrates the system response to a 5° slope applied between $t = 2$ s and $t = 10$ s. When the vehicle encounters the upward slope, its velocity decreases due to the additional resisting force from gravity. This climbing resistance acts against the vehicle's motion, causing a dip in speed. The proposed L-WSONM-based PIDD^2 controller compensates for this deviation; however, the vehicle speed recovers gradually, resulting in a small steady-state error. This behavior demonstrates the realistic performance of the controller, and adding feedforward compensation proportional to $Mg\sin(\theta)$

Table 14 Performance comparison between the proposed controller and recently developed PID controllers for cruise control systems

Methods	t_r (s)	t_s (s)	OS (%)	E_{ss} (km/h)
RPO-PID [33]	0.695	0.99989	0	0.021089
ObAOANM-PID [34]	0.746	1.172	0	0.007371
AOANM-PID [35]	0.803	1.257	0.41479	0.0012493
RSALF-PID [36]	0.692	1.072	0.35543	0.01092
MRFO–PID [35]	0.819	1.269	0.88861	0.0070882
L-WSONM-RPIDD^2 (Proposed)	0.463	0.692	0	0.00017022

could reduce the steady-state error and improve tracking performance under slope disturbances.

6.3 Sensitivity analysis to crosswind

This subsection assesses the performance of the L-WSONM-based RPIDD^2 controller in suppressing the effects of crosswind disturbances. Figure 18 presents the vehicle response to a crosswind disturbance. When a crosswind disturbance is applied, the vehicle velocity drops from 30 km/h to approximately 29.22 km/h. The proposed L-WSONM-based RPIDD^2 controller then acts to recover the vehicle speed back toward the reference value.

6.4 Comparison with recently developed controllers

In this section, we utilize the most effective methods reported in the literature to highlight the superior performance of the proposed L-WSONM-based RPIDD^2 controller for the cruise control system. For this purpose, we adopted PID controllers based on the red panda optimization algorithm (RPO) [33], opposition based learning arithmetic optimization with nelder–mead (ObAOANM) [34] the hybrid arithmetic optimization algorithm with nelder–mead (AOANM) [35], Lévy flight-based reptile search algorithm (RSALF) [36] and manta ray foraging algorithm (MRFO) [35].

As shown in Fig. 19, the comparative step responses confirm the strong performance of the proposed L-WSONM-based RPIDD^2 approach. The figure highlights the proposed approach's enhanced transient response in adapting to velocity variations within the cruise control system. Furthermore, detailed analysis of several time-domain metrics was undertaken to provide a clearer understanding of the proposed method's performance. The proposed controller demonstrates better performance than other optimized PID controllers in key time-domain measures; settling period, rise time, overshoot and steady-state error as detailed in Table 14. The findings demonstrate that the proposed method significantly enhances control precision and operational efficiency in cruise control systems, surpassing the performance of optimized PID controllers.

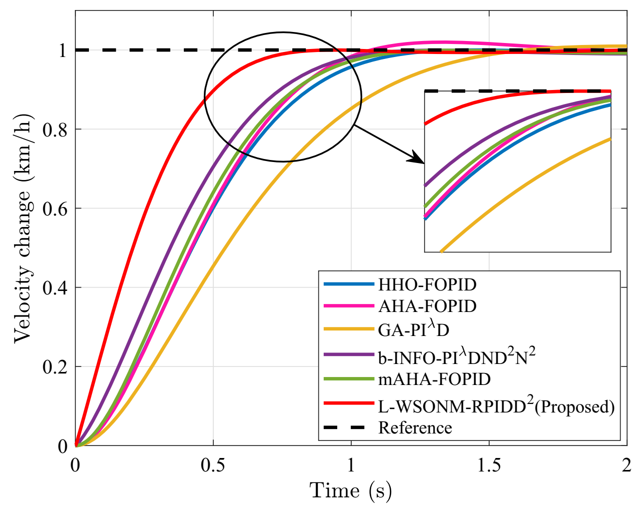

Fig. 20 Comparison of the closed-loop system's step responses with different optimized fractional-order controllers and the proposed approach

Next, to validate the efficacy of the proposed approach, it is rigorously compared with several fractional-order controllers, including FOPID, $PI^\lambda D$ and the $PI^\lambda DND^2N^2$ controller, as reported in the literature. These controllers tuned using Harris hawks optimization (HHO) [20], the artificial hummingbird algorithm (AHA) [37], the genetic algorithm (GA) [38], the boosted weighted mean of vectors (b-INFO) [39] and the modified artificial hummingbird algorithm (mAHA) [37]. This detailed comparison forms a solid basis for emphasizing the effectiveness of the proposed controller. The superiority of the proposed method becomes apparent when benchmarked against existing techniques, as shown in Fig. 20. As depicted in the figure, our approach demonstrates excellent effectiveness in delivering exceptional control capabilities. The corresponding time-domain performance metrics are provided in Table 15 to gain a clearer perspective on the operational dynamics of the proposed method. With respect to percent overshoot, rise time, settling time, and steady-state error, the b-INFO-$PI^\lambda DND^2N^2$ controller provides good performance metrics. However, the proposed method demonstrates superior performance compared to all optimized fractional-order controllers, including the b-INFO-$PI^\lambda DND^2N^2$ controller. The exceptional per-

Table 15 Performance comparison between the proposed controller and recently developed fractional-order controllers for the cruise control system

Methods	t_r (s)	t_s (s)	OS (%)	E_{ss} (km/h)
HHO-FOPID [20]	0.703	1.0722	0	0.0063141
AHA-FOPID [37]	0.658	0.986	1.9747	0.0029527
GA-$PI^{\lambda}D$ [38]	0.692	1.072	0	0.01092
b-INFO-$PI^{\lambda}DND^2N^2$ [39]	0.652	0.991	0	0.0096876
mAHA-FOPID [37]	0.661	1.043	0	0.0075974
L-WSONM-RPIDD^2 (Proposed)	0.463	0.692	0	0.00017022

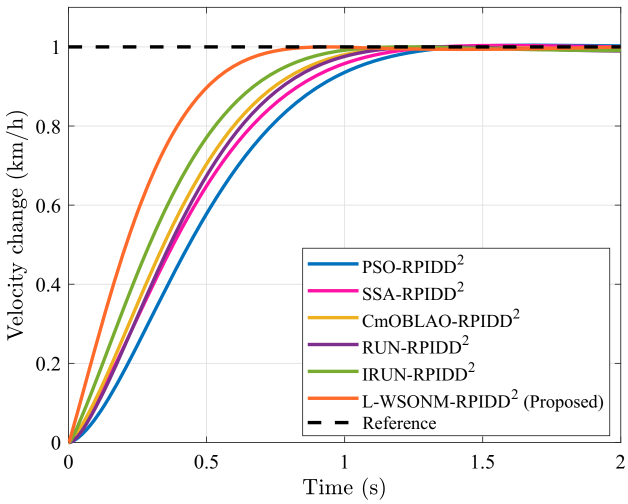

Fig. 21 Comparison of the closed-loop system's step responses with different optimized RPIDD^2 controllers and the proposed approach

formance of the proposed controller highlights its strong potential to improve control precision and system efficiency. These findings reinforce the method's effectiveness, marking it as a notable advancement in vehicle cruise control technology.

Finally, to thoroughly validate the potential of the proposed approach, extensive comparisons were carried out against RPIDD^2 controllers optimized using various techniques reported in the literature. This comprehensive evaluation included benchmarking the proposed method against RPIDD^2 controllers tuned using different algorithms, namely the particle swarm optimization (PSO) [22], the salp swarm algorithm (SSA) [22], and the Aquila Optimizer enhanced with a modified oppositional chaotic local search strategy (CmOBLAO) [22], Runge Kutta optimizer (RUN) [3] and its improved Runge Kutta optimizer (IRUN) [3]. These methods demonstrated effective and strong performance, making them a reliable benchmark for evaluating the strengths of our proposed approach. Figure 21 visually illustrates the superiority of the proposed L-WSONM-RPIDD^2 approach compared to other methods through comparative step response analysis. As shown in the figure, our proposed method significantly improves the system's transient response to velocity changes. Table 16 provides time-domain characteristics for various optimized RPIDD^2 controllers, offering a comprehensive evaluation of the developed controller. The L-WSONM-RPIDD^2 controller offers improved transient performance–faster rise time, faster settling time, and reduced overshoot–at the cost of a slightly higher steady-state error relative to the SSA-RPIDD^2 controller.

Table 17 presents the robustness analysis across eight operating scenarios and reveals clear performance differences among the RPIDD^2-based controllers. Overall, the L-WSONM-RPIDD^2 controller consistently provides the most favorable results in both transient and steady-state metrics. Across all scenarios, L-WSONM-RPIDD^2 achieves the lowest rise time (t_r) and settling time (t_s), demonstrating faster transient response, especially under variations in mass, aerodynamic drag coefficient, and initial velocity. The steady-state error (E_{ss}) results further highlight its superiority, as it yields the smallest E_{ss} in every scenario, often with a significant margin. Although SSA-RPIDD^2 occasionally offers slightly better overshoot values, it does not match

Table 16 Performance comparison between the proposed controller and recently developed RPIDD^2 controllers for the cruise control system

Methods	t_r (s)	t_s (s)	OS (%)	E_{ss} (km/h)
PSO-RPIDD^2 [22]	0.771	1.212	0.4229	0.0024781
SSA-RPIDD^2 [22]	0.732	1.112	0.35409	**0.00016423**
CmOBLAO-RPIDD^2 [22]	0.654	0.998	0	0.0039592
RUN-RPIDD^2[3]	0.668	1.022	0	0.010551
IRUN-RPIDD^2 [3]	0.602	0.909	0	0.0091911
L-WSONM-RPIDD^2 (Proposed)	**0.463**	**0.692**	**0**	0.00017022

The bold entries indicate the best (optimal) performance values among the compared methods

Table 17 Robustness analysis in the time domain under varying operational settings

Scenarios	Metrics	PSO-RPIDD^2	SSA-RPIDD^2	CmOBLAO-RPIDD^2	RUN-RPIDD^2	IRUN-RPIDD^2	L-WSONM-RPIDD^2
Scenario 1 ($M = 1600$, $\tau = 0.18$)	t_r (s)	0.844	0.798	0.712	0.728	0.654	**0.499**
	t_s (s)	1.382	1.293	1.194	1.225	1.079	**0.828**
	OS (%)	0	0	0	0	0	0
	E_{ss}	0.0013256	0.0018115	0.0046747	0.011468	0.0091673	**9.1437e−05**
Scenario 2 ($M = 1550$, $\tau = 0.19$)	t_r (s)	0.806	0.763	0.681	0.696	0.627	**0.479**
	t_s (s)	1.2953	1.196	1.083	1.109	0.984	**0.750**
	OS (%)	0.079788	0	0	0	0	0
	E_{ss}	0.00079788	0.0006323	0.0041541	0.010825	0.0090349	**9.5304e−5**
Scenario 3 ($M = 1400$, $\tau = 0.22$)	t_r (s)	0.711	0.677	0.606	0.618	0.559	**0.433**
	t_s (s)	1.071	0.982	0.878	0.902	0.802	**0.614**
	OS (%)	1.7626	1.5535	1.4808	1.6881	**1.3649**	1.4618
	E_{ss}	0.0043235	0.00050655	0.0045826	0.01118	0.0099861	**0.00048865**
Scenario 4 ($M = 1350$, $\tau = 0.16$)	t_r (s)	0.713	0.684	0.607	0.618	0.562	**0.429**
	t_s (s)	1.1456	1.118	1.024	1.031	0.949	**0.755**
	OS (%)	0	0	0	0	0	0
	E_{ss}	0.003691	0.0045057	0.0062691	0.013252	0.010305	**0.0010315**
Scenario 5 ($C_a = 0.9520$)	t_r (s)	0.765	0.726	0.650	0.664	0.598	**0.461**
	t_s (s)	1.2156	1.091	0.980	1.005	0.895	**0.684**
	OS (%)	0.85541	0.73716	0.34507	0.39845	0.29402	**0.010578**
	E_{ss}	0.0068365	0.0040044	**0.00049796**	0.0068969	0.0061337	0.0020263
Scenario 6 ($C_a = 1.4280$)	t_r (s)	0.778	0.737	0.659	0.673	0.606	**0.465**
	t_s (s)	1.2084	1.136	1.018	1.043	0.927	**0.702**
	OS (%)	0	0	0	0	0	0
	E_{ss}	**0.0018436**	0.0036467	0.0073962	0.014179	0.012109	0.0023572
Scenario 7 ($v_0 = 20$ km/h)	t_r (s)	0.761	0.723	0.647	0.661	0.596	**0.459**
	t_s (s)	1.218	1.078	0.969	0.994	0.886	**0.678**
	OS (%)	1.1458	0.99418	0.57638	0.64115	0.49561	**0.011557**
	E_{ss}	0.009763	0.006581	**0.0018231**	0.0044454	0.0041257	0.0034962
Scenario 8 ($v_0 = 35$ km/h)	t_r (s)	0.777	0.736	0.658	0.672	0.605	**0.465**
	t_s (s)	1.209	1.132	1.015	1.039	0.924	**0.700**
	OS (%)	0.065096	0.036907	**0**	**0**	**0**	0.0078405
	E_{ss}	**0.0011259**	0.0030136	0.0068251	0.013576	0.011614	0.0019934

The bold entries indicate the best (optimal) performance values among the compared methods the steady-state precision of L-WSONM-RPIDD^2. Overall, the results confirm that L-WSONM-RPIDD^2 delivers the best balance of rapid response, minimal overshoot, and high steady-state accuracy across diverse operating conditions. The results underscore the efficacy of the developed method, demonstrating that the vehicle cruise control system improves control precision, system efficiency, and reliability.

7 Conclusions

This paper presents the Lévy flight-based White Shark Optimization (WSO) algorithm integrated with the Nelder–Mead simplex method, referred to as L-WSONM, applied to vehicle cruise control systems. The proposed L-WSONM was benchmarked against state-of-the-art algorithms–including GWO, POA, GJO, RSA, WSO, TSA, GA, PDO, and SOA–using the CEC2020 test suites. Statistical metrics and Wilcoxon signed-rank tests demonstrate that L-WSONM achieves competitive or superior performance in solving complex optimization problems, with the exception of functions F2 and F9.

The integration of L-WSONM with the RPIDD^2 controller significantly enhances both precision and efficiency in vehicle cruise control. Extensive experimental and analytical evaluations confirm that the proposed framework is robust, adaptable, and capable of maintaining consistent performance across diverse operating conditions. Compared to existing PID, FOPID, RPIDD^2, and other advanced control schemes, the L-WSONM-based RPIDD^2 controller demonstrates clear advantages in convergence behavior, transient response, and overall control quality. However, it is noted that the SSA-based RPIDD^2 controller achieves a slightly lower steady-state error in certain scenarios, highlighting a limitation of the proposed method in this specific aspect.

However, despite these benefits, L-WSONM incurs a modest increase in computational cost due to the additional Nelder–Mead local search and Lévy flight steps. Furthermore, in simpler or low-dimensional scenarios, the performance improvement over standard WSO or other conventional optimizers may be limited. These limitations highlight that while L-WSONM provides a powerful and effective optimization tool, its application should be considered alongside computational constraints and problem complexity.

Overall, the proposed approach demonstrates the potential to enhance driving safety and operational efficiency by improving control accuracy and stability. The findings provide valuable insights for future advancements in automotive control systems and metaheuristic-based optimization, encouraging further exploration of hybrid optimization strategies in practical engineering applications.

Author Contributions H.B. wrote the entire manuscript text and prepared all figures and tables.

Funding Not applicable.

Data Availability Data are available from the authors upon reasonable request.

Declarations

Code availability Code is available from the first author upon reasonable request.

Conflict of interest The authors declare no Conflict of interest.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Vasebi S, Hayeri YM, Saghiri AM (2023) A literature review of energy optimal adaptive cruise control algorithms. IEEE Access 11:13636–13646
2. Theodosis D, Karafyllis I, Papageorgiou M (2023) Cruise controllers for lane-free ring-roads based on control Lyapunov functions. J Franklin Inst 360:6131–6161
3. Izci D, Rizk-Allah RM, Ekinci S, Hussien AG (2023) Enhancing time-domain performance of vehicle cruise control system by using a multi-strategy improved RUN optimizer. Alex Eng J 80:609–622
4. Blondin MJ, Trovão JP (2019) Soft-computing techniques for cruise controller tuning for an off-road electric vehicle. IET Intell Transp Syst 9:196–205
5. Zhang R, Wu N, Wang Z, Li K, Song Z, Chang Z, Chen X, Yu F (2023) Constrained hybrid optimal model predictive control for intelligent electric vehicle adaptive cruise using energy storage management strategy. J Energy Storage 65:107383
6. Wu W, Zou D, Ou J, Hu L (2020) Adaptive cruise control strategy design with optimized active braking control algorithm. Math Probl Eng 2020:8382734
7. Pradhan R, Pati BB (2018) Optimal FOPID controller for an automobile cruise control system. Proc 2018 Int Conf Recent Innov Electr Electron Commun Eng (ICRIEECE). Kolkata, India, pp 1436–1440
8. Zhao J, Wang Z, Lv Y, Na J, Liu C, Zhao Z (2024) Data-driven learning for H_∞ control of adaptive cruise control systems. IEEE Trans Veh Technol
9. Huang JY, Peng X, Hu L, Cao D (2020) Study on the driving style adaptive vehicle longitudinal control strategy. IEEE/CAA J Autom Sin 7:1107–1115
10. Onieva E, Godoy J, Villagra J, Milanés V, Pérez J (2013) On-line learning of a fuzzy controller for a precise vehicle cruise control system. Expert Syst Appl 40:1046–1053
11. Selvaraj DC, Hegde S, Amati N, Deflorio F, Chiasserini CF (2023) A deep reinforcement learning approach for efficient, safe and comfortable driving. Appl Sci 13:5272
12. Jabari M, Ekinci S, Izci D, Zitar RA, Migdady H, Smerat A, Abualigah L (2025) An advanced PID tuning method for temperature control in electric furnaces using the artificial rabbits optimization algorithm. Int J Dyn Control 13(5):1–15

13. Jabari M, Nasab MA, Zand M, Tightiz L, Padmanaban S (2024) Efficient parameter extraction in PV solar modules with the diligent crow search algorithm. Discover Energy 4(1):35
14. Jabari M, Izci D, Ekinci S, Bajaj M, Blazek V, Prokop L (2025) Hybrid adaptive ant lion optimization with traditional controllers for driving and controlling switched reluctance motors to enhance performance. Sci Rep 15:12898
15. Izci D, Ekinci S, Jabari M, Bajaj M, Blazek V, Prokop L, Mirjalili S (2025) A new intelligent control strategy for CSTH temperature regulation based on the starfish optimization algorithm. Sci Rep 15:12327
16. Pradhan R, Majhi SK, Pradhan JK, Pati BB (2018) Antlion optimizer tuned PID controller based on Bode ideal transfer function for automobile cruise control system. J Ind Inf Integr 9:45–52
17. Chaturvedi S, Kumar N (2023) Design and implementation of an optimized PID controller for the adaptive cruise control system. IETE J Res 69:7084–7091
18. Izci D, Ekinci S (2022) A novel hybrid ASO-NM algorithm and its application to automobile cruise control system. In: Proc 2nd Int Conf Artif Intell: Adv Appl (ICAIAA), Singapore, pp 333–343
19. Dawood SY, Mahmood AK, Ibrahim AM (2018) Comparison of PID, GA and fuzzy logic controllers for cruise control system. Int J Comput Digit Syst 7:311–319
20. Izci D, Ekinci S (2021) An efficient FOPID controller design for vehicle cruise control system using HHO algorithm. Proc 3rd Int Congr Hum-Comput Interact, Optim Rob Appl (HORA). Ankara, Turkey, pp 1–5
21. Sahib MA (2015) A novel optimal PID plus second order derivative controller for AVR system. Eng Sci Technol Int J 18:194–206
22. Ekinci S, Izci D, Abualigah L, Zitar RA (2023) A modified oppositional chaotic local search strategy based aquila optimizer to design an effective controller for vehicle cruise control system. J Bionic Eng 20:1828–1851
23. Ekinci S, Izci D, Kayri M (2022) An effective controller design approach for magnetic levitation system using novel improved manta ray foraging optimization. Arab J Sci Eng 47(8):9673–9694
24. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82
25. Braik M, Hammouri A, Atwan J, Al-Betar MA, Awadallah MA (2022) White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowl-Based Syst 243:108457
26. Farhat M, Kamel S, Elseify MA, Abdelaziz AY (2024) A modified white shark optimizer for optimal power flow considering uncertainty of renewable energy sources. Sci Rep 14:3051
27. Bencherqui A, Tahiri MA, El Ghouate N, Karmouni H, Sayyouri M, Askar SS, Abouhawwash M (2025) Chaos-enhanced white shark optimization algorithms (CWSO) for global optimization. Alex Eng J 122:465–483
28. Abd El-Sattar H, Kamel S, Elseify MA (2025) A modified white shark optimizer for optimizing photovoltaic, wind turbines, biomass, and hydrogen storage hybrid systems. J Energy Storage 113:115655
29. Wang ML, Gao BT, Lei JX, Quan XJ, Guan YF (2025) Improved white shark optimizer based maximum power point tracking algorithm for photovoltaic systems under partial shading conditions. J Electr Eng Technol 20:1293–1306
30. Al-Shalabi M, Mahdi MA, Braik M, Al-Betar MA, Ahamad S, Saad SA (2025) Opposition-based white shark optimizer for optimizing modified EfficientNetV2 in road crack classification. IEEE Open J Comput Soc
31. Houssein EH, Saeed MK, Al-Sayed MM (2024) EWSO: boosting white shark optimizer for solving engineering design and combinatorial problems. Math Comput Simul 225:1124–1153
32. Fathy A, Yousri D, Alharbi AG, Abdelkareem MA (2023) A new hybrid white shark and whale optimization approach for estimating the Li-ion battery model parameters. Sustainability 15:5667
33. Saravanan G, Pazhanimuthu C, Sathish Kumar D, Lalitha B, Senthilkumar M, Kannan E (2024) Red panda optimization algorithm-based PID controller design for automobile cruise control system. Proc Int Conf Smart Syst Electr. Electron, Commun Comput Eng (ICSSEECC), pp 33–37
34. Izci D, Ekinci S, Kayri M, Eker E (2021) A novel enhanced metaheuristic algorithm for automobile cruise control system. Electrica 21:1–10
35. Izci D, Ekinci S, Kayri M, Eker E (2022) A novel improved arithmetic optimization algorithm for optimal design of PID controlled and Bode's ideal transfer function based automobile cruise control system. Evolv Syst 13:453–468
36. Ekinci S, Izci D (2023) Enhanced reptile search algorithm with Lévy flight for vehicle cruise control system design. Evol Intell 16:1339–1351
37. Abualigah L, Ekinci S, Izci D, Zitar R (2023) Modified elite opposition-based artificial hummingbird algorithm for designing FOPID controlled cruise control system. Intell Autom Soft Comput 38:1–20
38. Idir A, Zemmit A, Khettab K, Nesri M, Guedida S, Canale L (2025) Performance analysis of fractionalized order PID controller based on metaheuristic optimization algorithms for vehicle cruise control systems. Sci Eng Technol 1:1–15
39. Ekinci S, Izci D, Abualigah L, Hussien AG, Thanh CL, Khatir S (2023) Revolutionizing vehicle cruise control: an elite opposition-based pattern search mechanism augmented INFO algorithm for enhanced controller design. Int J Comput Intell Syst 16:129
40. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
41. Al-Baik O, Alomari S, Alssayed O, Gochhait S, Leonova I, Dutta U, Malik OP, Montazeri Z, Dehghani M (2024) Pufferfish optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. Biomimetics 9:65
42. Chopra N, Ansari MM (2022) Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. Expert Syst Appl 198:116924
43. Abualigah L, Abd Elaziz M, Sumari P, Geem ZW, Gandomi AH (2022) Reptile search algorithm (RSA): a nature-inspired metaheuristic optimizer. Expert Syst Appl 191:116158
44. Kaur S, Awasthi LK, Sangal AL, Dhiman G (2020) Tunicate Swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Eng Appl Artif Intell 90:103541
45. Mirjalili S (2018) Genetic algorithm. In: Evolutionary algorithms and neural networks: theory and applications. Springer, pp 43–55
46. Ezugwu AE, Agushaka JO, Abualigah L, Mirjalili S, Gandomi AH (2022) Prairie dog optimization algorithm. Neural Comput Appl 34:20017–20065
47. Dhiman G, Kumar V (2019) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. Knowl-Based Syst 165:169–196
48. Yang XS, Deb S (2013) Multiobjective cuckoo search for design optimization. Comput Oper Res 40:1616–1624
49. Zhang Y, Jin Z, Zhao X, Yang Q (2020) Backtracking search algorithm with Lévy flight for estimating parameters of photovoltaic models. Energy Convers Manag 208:112615
50. Mokeddem D (2021) Parameter extraction of solar photovoltaic models using enhanced levy flight based grasshopper optimization algorithm. J Electr Eng Technol 16:171–179
51. Ekinci S, Izci D, Abu Zitar R, Alsoud AR, Abualigah L (2022) Development of Lévy flight-based reptile search algorithm with local search ability for power systems engineering design problems. Neural Comput Appl 34:20263–20283

52. Ekinci S, Izci D (2023) Enhanced reptile search algorithm with Lévy flight for vehicle cruise control system design. Evol Intell 16:1339–1351
53. Nelder JA (1965) A simplex method for function minimization. Comput J 7:308–313
54. Zhang H, Heidari AA, Wang M, Zhang L, Chen H, Li C (2020) Orthogonal Nelder-Mead moth flame method for parameters identification of photovoltaic modules. Energy Convers Manag 211:112764
55. Izci D, Hekimoğlu B, Ekinci S (2022) A new artificial ecosystem-based optimization integrated with Nelder-Mead method for PID controller design of buck converter. Alexandria Eng J 61:2030–2044
56. Ekinci S, Izci D, Abu Zitar R, Alsoud AR, Abualigah L (2022) Development of Lévy flight-based reptile search algorithm with local search ability for power systems engineering design problems. Neural Comput Appl 34:20263-20283
57. Izci D (2021) Design and application of an optimally tuned PID controller for DC motor speed regulation via a novel hybrid Lévy flight distribution and Nelder-Mead algorithm. Trans Inst Meas Control 43:3195–3211
58. Chen Y, Peng T, Kumar V, Sahoo N, Liu J, Ma Z, Qian Y, An H, Brown P (2019) Problem definitions and evaluation criteria for the CEC2020 special session and competition on single-objective bound-constrained numerical optimization. Zhengzhou Univ, Zhengzhou, China, Tech Rep, Comput Intell Lab, p 201911