Alexandria Engineering Journal 80 (2023) 609–622

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original Article

Enhancing time-domain performance of vehicle cruise control system by using a multi-strategy improved RUN optimizer

Davut Izci [a,b], Rizk M. Rizk-Allah [c], Serdar Ekinci [a], Abdelazim G. Hussien [d,e,*]

[a] Department of Computer Engineering, Batman University, Batman 72100, Turkey
[b] MEU Research Unit, Middle East University, Amman, Jordan
[c] Basic Engineering Science Department, Faculty of Engineering, Menoufia University, Shebin El-Kom 32511, Egypt
[d] Department of Computer and Information Science, Linköping University, Linköping, Sweden
[e] Faculty of Science, Fayoum University, Fayoum, Egypt

ARTICLE INFO

Keywords:
Cruise control system
Master/slave approach
Runge Kutta optimizing algorithm
Controller design

ABSTRACT

This paper addresses the pressing concern of traffic safety by focusing on the optimization of vehicle cruise control systems. While traditional control techniques have been widely employed, their design procedures can be time-consuming and suboptimal. To overcome these limitations, metaheuristic algorithms have been introduced as promising solutions for complex optimization problems. In this study, an improved Runge Kutta optimizer (IRUN) is developed and applied to enhance the control performance of a real PID plus second-order derivative (RPIDD^2) controller for vehicle cruise control systems. The IRUN optimizer incorporates advanced strategies such as quadratic interpolation, Laplacian segment mutation, Levy flight, and information-sharing-based local search mechanisms. By integrating these strategies, the IRUN algorithm demonstrates enhanced optimization capabilities, making it well-suited for tuning the controller. The proposed approach utilizes a master–slave system, where the ideal reference model sets the desired response and the RPIDD^2 controller adjusts its parameters accordingly. The integral of the square error is employed as the objective function to evaluate the control system's performance. Statistical analyses, convergence analyses, and stability evaluations and robustness analysis are performed to demonstrate the effectiveness of the IRUN-based RPIDD^2 controller. Comparative studies are conducted against established approaches using PID, fractional-order PID (FOPID), and RPIDD^2 controllers, showcasing the superiority and effectiveness of the proposed approach. Overall, this paper presents a comprehensive study on enhancing the time-domain performance and stability of vehicle cruise control systems, providing significant improvements in control accuracy and efficiency. The subsequent sections delve into the proposed approach, experimental setup, and obtained results, further emphasizing the significance and potential impact of this research.

1. Introduction

Vehicles have undoubtedly brought immense convenience to the public, but they also give rise to a range of significant social issues, with traffic safety being a particularly pressing concern. Research estimates indicate that approximately 90% of traffic accidents stem from human error, including factors such as fatigue, impaired driving, distractions, misjudgment, and a lack of safety awareness [1]. In recent years, substantial efforts have been directed towards mitigating the occurrence of traffic accidents, with a focus on technologies that can assist drivers, such as advanced driver assistance systems [2] and autonomous driving systems [3]. Among these technologies, cruise control stands out as one of the most prominent and widely utilized [4].

In the realm of automotive engineering, vehicle cruise control systems play a crucial role in enhancing driving comfort, fuel efficiency, and overall road safety [5,6]. These systems are designed to maintain a desired speed for the vehicle, relieving the driver from constantly adjusting the throttle and allowing them to focus on other critical aspects of driving [7,8]. In that sense, the accurate and efficient control of vehicle cruise control systems is of utmost importance. Traditional control techniques, such as proportional-integral-derivative (PID) controllers, have been widely employed in cruise control systems [9–11].

* Corresponding author at: Department of Computer and Information Science, Linköping University, Linköping, Sweden.
E-mail addresses: davut.izci@batman.edu.tr, davutizci@gmail.com (D. Izci), rizk_masoud@yahoo.com (R.M. Rizk-Allah), serdar.ekinci@batman.edu.tr (S. Ekinci), abdelazim.hussien@liu.se (A.G. Hussien).

https://doi.org/10.1016/j.aej.2023.09.009
Received 24 June 2023; Received in revised form 15 August 2023; Accepted 1 September 2023
Available online 8 September 2023
1110-0168/© 2023 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Engineering, Alexandria University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Apart from PID controllers, different structures, including fractional-order PID (FOPID) [12], model predictive [13,14], deep reinforcement learning [15], fuzzy logic [16] and real PID plus second order derivative [17] controllers have also been explored in the literature for cruise control systems.

While these controllers offer satisfactory performance in many cases, their design procedure can be time-consuming and may not always yield optimal results. To address these limitations, researchers have turned to metaheuristic algorithms, which offer promising solutions for complex optimization problems [18]. In the context of vehicle cruise control systems, the application of metaheuristic algorithms can significantly enhance the time-domain performance and stability of the control system. Therefore, this paper focuses on the development and application of an improved Runge Kutta (RUN) optimizer [19], named IRUN, to achieve superior control performance. The proposed IRUN optimizer incorporates several advanced strategies to enhance its effectiveness. These strategies include quadratic interpolation, Laplacian segment mutation, Levy flight, and information-sharing-based local search mechanisms. By integrating these strategies into the optimizer, the resulting IRUN algorithm demonstrates enhanced optimization capabilities, making it well-suited for tuning the controller.

In this study, the real PID plus second-order derivative (RPIDD^2) controller is adopted as the control mechanism for the vehicle cruise control system as it is capable of improving the time-domain stability [20]. The aim is to optimize the RPIDD^2 controller's parameters using the IRUN optimizer, thus ensuring accurate and efficient control of the system. To further improve the performance, a master–slave approach is also employed, utilizing an ideal reference model [21]. The master system, represented by the ideal reference model, establishes a desired response for the cruise control system. The slave system, consisting of the RPIDD^2 controller, adjusts the system's parameters to match the desired response dictated by the ideal reference model. Throughout the experimentation process, the integral of the square error (ISE) performance metric is employed as the objective function. This metric provides a quantitative measure of the control system's performance, enabling a comprehensive evaluation of the proposed approach. By employing the smart optimization algorithm, IRUN, as the tuning mechanism for the RPIDD^2 controller, the overall proposed approach aims to enhance the time-domain performance and stability of the cruise control system.

In order to comprehensively evaluate the effectiveness of the proposed method, a rigorous assessment is conducted utilizing various statistical performance analyses, Wilcoxon's test, convergence analyses, and time-domain-based stability evaluations. These comprehensive analyses showcase the remarkable capabilities of the IRUN algorithm, surpassing the performance of the original version of the RUN optimizer. Furthermore, a comparative study is undertaken, pitting the proposed approach against established techniques employing PID, fractional-order PID (FOPID), and RPIDD^2 controllers. To substantiate the promise of the proposed approach, meticulous comparisons are made with PID controllers tuned using well-performing methods such as the antlion optimizer [22], enhanced reptile search algorithm with Lévy flight [23], and improved arithmetic optimization algorithm [11]. Furthermore, an in-depth investigation is carried out by evaluating the proposed approach against FOPID controllers fine-tuned through highly effective methodologies like Harris hawks optimization [24], artificial hummingbird algorithm [25], and modified artificial hummingbird algorithm [25]. Lastly, the superiority of the proposed approach is further established by comparing it with RPIDD^2 controllers tuned by prominent techniques such as particle swarm optimization [17], salp swarm algorithm [17], and modified oppositional chaotic local search strategy based aquila optimizer [17].

By benchmarking against these well-established methods, the proposed approach consistently exhibits its superiority and effectiveness in enhancing the performance of vehicle cruise control systems. This comprehensive study focuses on advancing the time-domain performance and stability of these systems through the utilization of the IRUN optimizer and a master–slave approach. The achieved improvements in control accuracy and efficiency are of significant magnitude, underscoring the significance and potential impact of this research. The subsequent sections of this paper delve into the intricate details of the proposed approach, encompassing the experimental setup and the compelling results obtained. Through this thorough exploration, the research not only solidifies its scientific contributions but also provides valuable insights for the broader field of automotive engineering. Overall, this work offers a comprehensive and innovative approach to enhance the time domain performance of vehicle cruise control systems, incorporating advanced optimization algorithms and methodologies. Through its contributions, the research provides valuable insights and advancements to the field of automotive engineering and control system design. Considering the abovementioned discussion, the major contributions for the presented work can be stated as follows:

- A novel IRUN optimizer, an improved version of the Runge Kutta (RUN) optimizer, is presented which incorporates advanced strategies including quadratic interpolation, Laplacian segment mutation, Levy flight, and information-sharing-based local search mechanisms for the first time.
- The IRUN embeds the above multi-strategy scheme to boost a balance among the explorative and exploitative capabilities with an exceptional ability to avoid local optima in terms of incorporation of enhancement strategy.
- A master–slave approach is introduced by utilizing an ideal reference model to guide the tuning process, thereby enhancing the control system's accuracy and efficiency.
- This novel IRUN optimizer together with the master–slave approach is employed to enhance the control performance of an RPIDD^2 controller for vehicle cruise control system for the first time.
- The use of the ISE as a performance metric adds a quantitative and objective measure to assess control system performance.
- The performance of the IRUN is rigorously validated and evaluated using PID, FOPID, and RPIDD^2 controllers in terms of verification and assessment of effectiveness. The work extensively evaluates the proposed approach through various statistical performance analyses, convergence analyses, stability evaluations, and comparative studies against well-established techniques. This rigorous assessment highlights the superiority of the proposed method across a range of control scenarios.
- The research's findings and methodology contribute to the broader field of automotive engineering by advancing control system optimization, with potential applications beyond vehicle cruise control systems.
- This study demonstrates substantial improvements in time-domain performance and stability, substantiating the potential impact of the proposed approach on real-world driving scenarios.

2. Original form of Runge Kutta optimizer

A novel *meta*-heuristic framework known as the RUNge–Kutta (RUN) optimizer, derived from the iterative Runge-Kutta (IRK) model, has been proposed by Ahmadianfar et al. [19]. The IRK model is commonly used to find approximate solutions for ordinary differential equations (ODEs). From an optimization perspective, the RUN optimizer begins by initializing M agents within the search landscape. It employs two operators, namely the Runge-Kutta-based searching mechanism (SM) and the solution enhancement operator (SEO), for updating the current solution (Δ_i). The current solution is updated based on the SM perspective of IRK theory as follows.

$$\Delta_{new1} = \begin{cases} (\Delta_b + r.SF.e.\Delta_b) + SF.SM + \omega.randn.(\Delta_d - \Delta_b) & rand < 0.5 \\ (\Delta_d + r.SF.e.\Delta_d) + SF.SM + \omega.randn.(\Delta_{a1} - \Delta_{a2}) & Otherwise \end{cases} \quad (1)$$

where r stands for an integer with 1 or −1 value, e denotes a random number ranged from 0 to 2, ω denotes a random number; Δ_{a1} and Δ_{a2} are two solutions that are selected at random in the range of [0, M]; M denotes the population size; *randn* defines a random number that generates values based on the normal distribution; SF stands for an adaptive factor that is expressed as follows:

$$SF = 2.(0.5 - rand) \times \delta \tag{2}$$

$$\delta = 10 \times exp\left(-12 \times rand \times \left(\frac{it}{T_{max}} \right) \right) \tag{3}$$

where it and T_{max} denote the present iteration and largest iteration number, respectively. Further, the Δ_b and Δ_d are represented by:

$$\Delta_b = \varphi \times \Delta_i + (1 - \varphi) \times \Delta_{a1}, i = 1, 2, .., M \tag{4}$$

$$\Delta_d = \varphi \times \Delta_{best} + (1 - \varphi) \times \Delta_{cbest} \tag{5}$$

where φ signifies a random number within the range (0,1), Δ_{best} identifies the best-so-far solution, and Δ_{cbest} represents the best position detected at each iteration. The SM is given by:

$$SM = \frac{1}{6}(\Delta_{IRK}).\Delta x \tag{6}$$

$$\Delta_{IRK} = g_1 + 2.g_2 + 2.g_3 + g_4$$

$$g_1 = \frac{1}{2\Delta x}(rand.\Delta_{\mathrm{ws}} - v.\Delta_{best})$$

$$v = (1 - rand).round(1 + rand)$$

$$g_2 = \frac{1}{2\Delta x}(rand.(\Delta_{\mathrm{ws}} + r_1.g_1.\Delta x) - (v.\Delta_{best} + r_2.g_1.\Delta x))$$

$$g_3 = \frac{1}{2\Delta x}(rand.(\Delta_{\mathrm{ws}} + r_1.\left(\frac{1}{2}g_2\right).\Delta x) - (v.\Delta_{best} + r_2.\left(\frac{1}{2}g_2\right).\Delta x))$$

$$g_4 = \frac{1}{2\Delta x}(rand.(\Delta_{\mathrm{ws}} + r_1.g_3.\Delta x) - (v.\Delta_{best} + r_2.g_3.\Delta x))$$

where Δ_{ws} defines the worst solution; and r_1 and r_2 define two random numbers generated within the range of [0, 1]. The step formula (Δx) is given by:

$$\Delta x = 2.rand.|Sp| \tag{7}$$

$$Sp = rand.\left(\left(\Delta_{best} - rand.\Delta_{avg}\right) + \alpha\right)$$

$$\alpha = rand.(\Delta_i - rand.(ub - lb)).exp(-4.\frac{it}{T_{max}})$$

where Sp defines the step size; lb and ub defines the lower and upper bounds for the candidate problem; Δ_{avg} stands for the average of solutions during the candidate iteration. Furthermore, Δ_{ws} and Δ_{best} are defined as follows.

$$iff(\Delta_i) < f(\Delta_{bs,i})$$

$$\Delta_{best} = \Delta_i$$

$$\Delta_{\mathrm{ws}} = \Delta_{bs,i}$$

$$else$$

$$\Delta_{best} = \Delta_{bs,i}$$

$$\Delta_{\mathrm{ws}} = \Delta_i$$

$$end \tag{8}$$

where $\Delta_{bs,i}$ defines the best solution among three randomly generated members (Δ_{a1}, Δ_{a2}, *and* Δ_{a3}).

To enhance the quality of solutions and avoid using local optima tramping, RUN makes use of an efficient operator, SEO. The RUN optimizer incorporates the SEO operator using the following scenario to create the solution Δ_{SEO}.

$$\Delta_{SEO} = \begin{cases} \Delta_{new2} + \lambda.\rho.\left|\left(\Delta_{new2} - \Delta_{avg}\right) + randn\right| & w < 1 \\ \left(\Delta_{new2} - \Delta_{avg}\right) + \lambda.\rho.\left|\left(2.rand.\Delta_{new2} - \Delta_{avg}\right) + randn\right| & otherwise \end{cases} \tag{9}$$

$$\rho = rand(0, 2).exp\left(-c\left(\frac{it}{T_{max}} \right) \right) \tag{10}$$

$$\Delta_{avg} = \frac{\Delta_{a1} + \Delta_{a2} + \Delta_{a3}}{3} \tag{11}$$

$$\Delta_{new2} = \gamma \times \Delta_{avg} + (1 - \gamma) \times \Delta_{best} \tag{12}$$

where γ denotes a random number generated within the range of [0, 1], λ stands for an integer number that takes 1, 0, or −1, and c signifies a random number that is set to 5 $\times$ *rand*.

Due to the possibility that solution (Δ_{SEO}) may not provide a better fitness function than of the current solution Δ_{new1} (i.e., $f(\Delta_{\mathrm{SEO}}) > f(\Delta_{new1})$), another hopes of coming up with a promising answer, solution Δ_{new3} is generated, which is considered as follows where $rand_1$, $rand_2$, and $rand_3$ define random values.

$$\Delta_{new3} = (\Delta_{SEO} - rand_1.\Delta_{SEO}) + SF.(rand_2.\Delta_{IRK} + (2.rand_3.\Delta_{best} - \Delta_{SEO})) \tag{13}$$

3. Recommended improvement strategy for RUN optimizer

Through careful analysis of the development of the RUN optimizer, it is evident that while it possesses advantages such as a simple structure and high precision, there are certain limitations that hinder its ability to effectively handle multimodal and complex problems [18,26]. These drawbacks primarily arise from the challenge of striking a balance between exploration and exploitation. Specifically, the sequential updating of search agents in the RUN optimizer, where weaker solutions are eliminated and better solutions survive, fails to take into account the valuable self-experience information of the population. Consequently, the updated members may lack the potential to explore promising regions. Furthermore, like many optimization approaches, the classical RUN is susceptible to stagnation, often getting trapped in local optima and failing to achieve improved outcomes [27]. To address these issues and enhance the efficiency of the RUN optimizer, we propose an enhanced approach called IRUN. The IRUN method incorporates several key strategies, including quadratic interpolation strategy (QIS), Laplacian segment mutation (LSM), Levy flight strategy (LFS), and information-sharing-based local search (ISLS). By integrating these techniques, IRUN aims to overcome the shortcomings of the classical RUN algorithm and achieve superior performance in terms of convergence and solution quality.

The QIS is integrated to enhance the exploitation feature while refining the solution accuracy of RUN optimizer. The QIS uses the approximate polynomial interpolation technique through using the knowledge of the given function at some candidate positions to formulate a low order interpolating polynomial that closely approximates the given objective function, and then use the polynomial's optimal solution

as a close approximation of the optimal solution of the given function [28]. Hence, a learning strategy using the function information can enhance the exploration ability. To be specific, the i^{th} individual and its two surrounding individuals $(i+1)^{th}$, and $(i+2)^{th}$ are selected, then by using the information of three individuals, a quadratic function can be expressed which is denoted by $Q(\Delta) = a_0 + a_1\Delta + a_2\Delta^2$, $a_0, a_1, a_2 \in R$. Let $Q(\Delta)$ represents the interpolating polynomial of the exact function $(f(\Delta))$ at abscissae Δ_i, Δ_{i+1}, Δ_{i+2}. That is.

$$\begin{cases} Q(\Delta_i) = a_0 + a_1\Delta_i + a_2\Delta_i^2 = f(\Delta_i) \\ Q(\Delta_{i+1}) = a_0 + a_1\Delta_{i+1} + a_2\Delta_{i+1}^2 = f(\Delta_{i+1}) \\ Q(\Delta_{i+2}) = a_0 + a_1\Delta_{i+2} + a_2\Delta_{i+2}^2 = f(\Delta_{i+2}) \end{cases} \tag{14}$$

The minimal value of the approximate quadratic function $Q(\Delta)$ can be attained at

$$\Delta^* = \frac{1}{2} \times \frac{(\Delta_{i+1}^2 - \Delta_{i+2}^2)f(\Delta_i) + (\Delta_{i+2}^2 - \Delta_i^2)f(\Delta_{i+1}) + (\Delta_i^2 - \Delta_{i+1}^2)f(\Delta_{i+2})}{(\Delta_{i+1} - \Delta_{i+2})f(\Delta_i) + (\Delta_{i+2} - \Delta_i)f(\Delta_{i+1}) + (\Delta_i - \Delta_{i+1})f(\Delta_{i+2})} \tag{15}$$

By this strategy, an improved solution can be obtained with better fitness value than the previous one. Furthermore, the updating process can be performed using a greedy manner as follows.

$$\begin{cases} \Delta_i^{it+1} = \Delta_i^{it} & f(\Delta_i^{it}) \le f(\Delta_{best}) \\ \Delta_i^{it+1} = \Delta_{best} & f(\Delta_i^{it}) \rangle f(\Delta_{best}) \end{cases} \tag{16}$$

where *it* represents the current iteration of algorithm. After applying the QIS to each individual, the quality of the population can be enriched.

In IRUN algorithm, the individuals are behaved using some random trends during the SM and SEO stages. This scheme of updating may lead to infective diversification pattern. Therefore, the LSM is adopted at aiming to improve solution' diversity and hence boost the exploration feature [29]. The LSM method, rooted in the Laplacian distribution strategy, is designed to enhance solution diversity while also leveraging information from the current solution and the best so far solution to carry out the updating process. The primary objective of LSM is indeed to improve solution diversity by utilizing the Laplacian distribution's property of encouraging exploration in the solution space. Additionally, by incorporating information from the best solution, LSM aims to strike a balance between exploration and exploitation. This results in a diverse set of solutions that still benefit from the guidance of promising regions. In particular, the LSM is initiated with two individuals, the one of them is selected randomly (Δ_a) and the other one represents the best solution obtained so for (Δ_{best}). Therefore, According to Laplacian distribution, the most probable search pattern is as follows.

$$L_j = \begin{cases} \beta - \gamma log_e(u_j) & v_j \le 0.5 \\ \beta + \gamma log_e(u_j) & v_j > 0.5 \end{cases} \tag{17}$$

where u_j, v_j stand for two arbitrary numbers generated in the interval $[0, 1]$ using uniform distribution. Here, $\beta \in R$ *and* γ define the location factor and the scaling coefficient, respectively. Thus, the renewed position of the individua can be formulated as follows.

$$\Delta_1^{LSM} = \Delta_a + L \times |\Delta_a - \Delta_{best}|$$

$$\Delta_2^{LSM} = \Delta_{best} + L \times |\Delta_a - \Delta_{best}| \tag{18}$$

where Δ_1^{LSM} and Δ_1^{LSM} define the new solutions obtained based on LSM. In this regard, LSM establishes fresh solutions in the vicinity of their original ones for a small value of γ, and establishes fresh solutions far from their original ones for the large value of γ. When β and γ are fixed values, the LSM can distribute the new solutions proportional to the parents' distribution.

To mitigate the loss of diversity with the growth of population and enhance the randomness of the population during the iterative process, the Levy flight strategy (LFS) is adopted as the random walk method based on Levy distribution [30]. LSF takes a different approach to maintaining population diversity. It exploits the neighborhood of the current solution using the Levy flight concept. By leveraging the Levy flight's ability to explore a wide range of solution space, LSF encourages solutions to venture into distant and potentially unexplored regions. This approach indirectly contributes to diversity preservation by allowing solutions to escape local optima and explore a broader range of possibilities. Thus, LFS is presented to exploit near the solution with promising search ability. According to the LFS, some new solutions can be generated to eliminate the others with poor search ability with the aim to maintain the population 'diversity as well as enable the population to jump out of the local optima. Therefore, the updating process using LFS can be expressed as follows.

$$\Delta_i^{LFS} = \Delta_i.(1 + levy(\beta)) \tag{19}$$

$$\Delta_i = \begin{cases} \Delta_i^{LFS} & f(\Delta_i^{LFS}) \langle f(\Delta_i) \\ \Delta_i & otherwise \end{cases} \tag{20}$$

where Δ_i^{LFS} defines the renewed position using the LFS, Δ_i defines the present solution, and $levy(\beta)$ denotes the Lévy flight terms that will be illustrated introduced next. Eq. (20) signifies that the present solution is substituted with renewed one based on levy flight perspective if it is better; otherwise, present one is retained. The lévy fly step might increase the strength of the algorithm as it is characterized by small and larger jumps steps which improve the exploration and exploitation abilities. From the perspective of implementation, the Mantegna method is used to determine the step size of the Lévy flight [31] as follows:

$$levy(\beta) = \left(\frac{\sigma.u}{|v|^{(1/\beta)}} \right) \tag{21}$$

where u and v represent two parameters drawn from Gaussian distributions as $u\ N(0,1)$ and $v\ N(0,1)$ where β defines a distribution index and σ defines the standard deviation that is expressed as follows where $\Gamma()$ stands for Gamma function.

$$\sigma = \left\{ \frac{\Gamma(1+\beta)sin(\frac{\pi\beta}{2})}{\beta.\Gamma(\frac{1+\beta}{2}).2^{\frac{(\beta-1)}{2}}} \right\}^{1/\beta} \tag{22}$$

In the traditional RUN, the local search is carried out in SEO stage using average solution of all solutions. Despite the fact that this update is a useful search tool in some tasks, the solutions are easily trapped at the local optimum challenging with the worse location the population's average.

To remedy this shortcoming and strengthen the efficiency of local search, the ISLS is developed to strengthen the exploitation feature through sharing the information of the best leader (Δ^{P}) and worst solution (Δ^{W}) of the elite group within the population. To be specific, the elite group of chosen with identifying the information of Δ_{best}, Δ^{P} and Δ^{W}. Afterwards, the updating process is presented using three types of movements, which are moving Δ^{W} towards Δ^{P}, or moving Δ^{W} towards Δ_{best}, and moving Δ^{W} towards the average of Δ^{P} and Δ_{best}. The movements are performed sequentially and it is stopped when the improved fitness comes to a specific one. The updating process of this stage is as

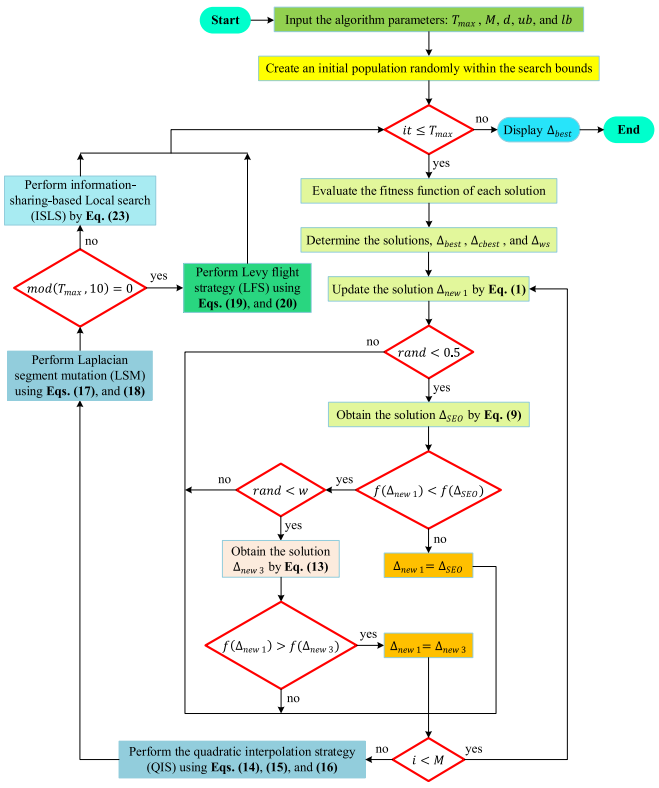


Fig. 1. Flowchart of developed IRUN algorithm.

follows:

$$\Delta_I^{it+1} = \begin{cases} \Delta_1^{ISLS} = 2 \times a_1 \times \left(\Delta^P - \Delta^W\right) + \Delta^W & iff\,(\Delta_1^{ISLS}) \le f(\Delta_I^{it}) \\ \Delta_2^{ISLS} = 2 \times a_2 \times \left(\Delta_{best} - \Delta^W\right) + \Delta^W & elseiff\,(\Delta_2^{ISLS}) \le f(\Delta_I^{it}) \\ \Delta_3^{ISLS} = 2 \times a_3 \times \left(\left(\Delta^P + \Delta_{best}\right)/2 - \Delta^W\right) + \Delta^W & otherwise \end{cases} \tag{23}$$

where Δ_I^{it} and Δ_I^{it+1} defines the current and updated solutions of the I^{th} individual within the elite group. The flowchart of the proposed IRUN framework is illustrated in Fig. 1.

4. Mathematical model of linearized vehicle cruise control system

The vehicle cruise control system plays a crucial role in maintaining the desired speed for an automobile, providing convenience, and enhancing driving comfort. The dynamic model of a vehicle cruise control system is illustrated in Fig. 2. In order to design and analyze the control system, it is essential to develop an accurate mathematical model that captures the dynamics of the vehicle. Table 1 outlines the parameter values used for the test system, which are crucial in defining the characteristics and behavior of the vehicle.

The aerodynamic drag coefficient represents the resistance experienced by the vehicle due to air drag. The actuator constant characterizes the responsiveness and efficiency of the control actuator. The observation time denotes the sampling time interval at which the control system measures and processes information. The driver's reaction time refers to the delay between perceiving a change in the vehicle's state and initiating a control action. The mass of the vehicle and passenger affects the overall dynamics and inertia of the system. Gravitational acceleration is the acceleration due to gravity, which influences the vehicle's behavior on inclined surfaces. The minimum and maximum saturation limits define the range within which the control signals can vary. By considering these parameters (and some of the assumptions outlined in [17]),

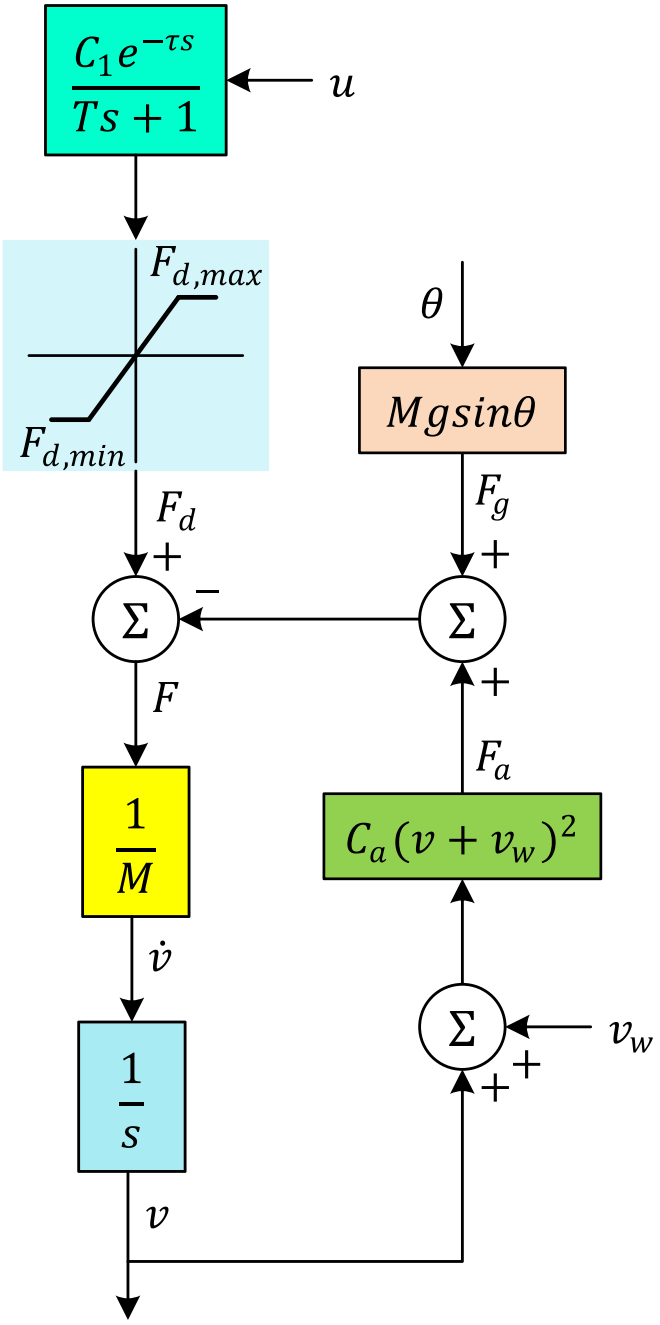


Fig. 2. Dynamic model of a vehicle cruise control system.

Table 1
The parameter values used for test system.

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

we can obtain the transfer function that represents the linearized vehicle cruise control system. The transfer function, denoted as $G_{plant}(s)$, can be can be obtained for the linearized model of the vehicle cruise control system where $C = \frac{C_1}{MT_\tau}$, $p_1 = 2\frac{C_a v_0}{M}$, $p_2 = \frac{1}{T}$ and $p_3 = \frac{1}{\tau}$.

$$G_{plant}(s) = \frac{C}{(s+p_1)(s+p_2)(s+p_3)} = \frac{2.4767}{(s+0.0476)(s+1)(s+5)} \tag{24}$$

The obtained transfer function, derived using the values presented in Table 1, captures the dynamics and behavior of the linearized vehicle cruise control system. It allows for further analysis and design of control strategies to achieve desired speed control and enhance driving experience.

5. Proposed approach for enhancing performance of the vehicle cruise control system

5.1. RPIDD^2 controller

The conventional PID controller is characterized by its three main gains: the proportional gain (K_p), integral gain (K_i), and derivative gain (K_d). These gains determine the contributions of the proportional, integral, and derivative components in the controller's output and are described with the following form [32].

$$C_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \tag{25}$$

The traditional PID controller finds extensive use across various fields [33]. However, when it comes to dynamic systems such as cruise control systems, a more flexible controller is desired. This has led to the exploration of different variations of the PID controller. In line with this, the concept of employing a fractional-order controller for dynamic systems emerged, giving rise to the development of the fractional order PID (FOPID) controller [34]. The FOPID controller represents an enhanced version that introduces fractional orders (λ and μ) into its structure. Its transfer function can be defined as follows:

$$C_{FOPID}(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \tag{26}$$

where K_p, K_i, and K_d are proportional, integral, and derivative gains, while λ and μ stand for fractional integral and derivative orders, respectively. A more recent variant of the PID controller is the PIDD^2 controller, which has demonstrated improvements in system stability and steady-state accuracy [35]. The PIDD^2 controller is obtained by introducing a second-order derivative term to the classical PID controller. This addition reduces steady state error, and contributes to overall system stability. The transfer function of the traditional PIDD^2 controller can be expressed as follows.

$$C_{ideal-PIDD^2}(s) = K_p + \frac{K_i}{s} + K_d s + K_{dd} s^2 \tag{27}$$

In the above equation, the derivative term is less effective in the high-frequency domain. This is due to the amplification of control signal by the derivative term in the presence of sensor noise, which can negatively impact the performance of the control system. To address this issue, a low-pass filter can be incorporated into the derivative term [36]. This modification gives rise to the real PIDD^2 (RPIDD^2) controller, which is expressed as follows:

$$C_{RPIDD^2}(s) = K_p + \frac{K_i}{s} + K_d\left(\frac{N_1 s}{s+N_1}\right) + K_{dd}\left(\frac{N_2 s}{s+N_2}\right)^2 \tag{28}$$

where K_p, K_i, K_d, and K_{dd} denote proportional, integral, derivative, and second-order derivative gains, respectively. N_1 and N_2 represent the filter coefficients. The block diagram shown in Fig. 3 presents an RPIDD^2 controlled vehicle cruise control system.

5.2. Master-slave approach

In this study, a sophisticated approach known as the master–slave method [21] is employed to improve the performance of the cruise control system. The idea behind this approach is to have a master system, called the ideal reference model, developed by H.W. Bode [37], that sets a desired response for the cruise control system. The goal is for the cruise control system to follow this reference model accurately. To achieve this, a smart optimization algorithm called IRUN is used to tune an RPIDD^2 controller, which acts as the slave system. The RPIDD^2 controller is responsible for adjusting the cruise control system's parameters to match the desired response dictated by the ideal reference model.

Using the ideal reference model as the master system offers significant advantages, which have been recognized in various studies [11,22,38–44]. The ideal reference model is described by the following equation that represents an ideal open loop transfer function.

$$L(s) = \left(\frac{\omega_c}{s}\right)^\alpha \tag{29}$$

It has two parameters: ω_c, which determines the frequency at which the response transitions from gain to phase, and α, a real number that influences the shape of the magnitude and phase curves. In this study, ω_c is set to 2.5, and α is chosen as 1. This means that the ideal reference model exhibits a consistent phase margin regardless of changes in gain, maintaining stability while only adjusting the crossover frequency. Eq. (30) presents the ideal closed-loop transfer function model, denoted as $TF(s)$, with unit feedback. The ideal reference model serves as the master system, guiding the cruise control system, while the IRUN algorithm based RPIDD^2 controller acts as the slave system, ensuring that the cruise control system faithfully follows the desired response dictated by the ideal reference model.

$$CLTF(s) = \frac{L(s)}{1+L(s)} = \frac{\omega_c^\alpha}{s^\alpha + \omega_c^\alpha} \tag{30}$$

Overall, this study employs a sophisticated master–slave approach, with the ideal reference model as the master and the IRUN algorithm based RPIDD^2 controller as the slave, to enhance the performance of the cruise control system and achieve precise adherence to a predefined reference model.

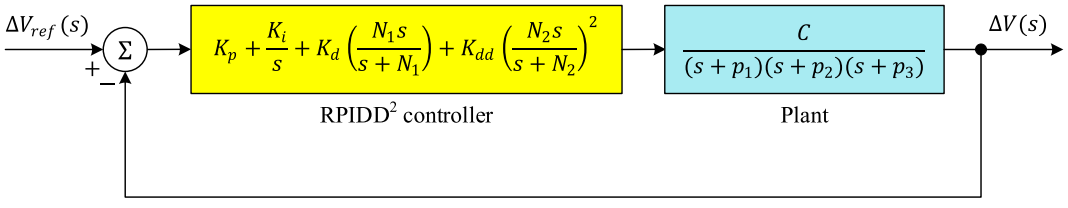


Fig. 3. Block diagram of vehicle cruise control system with RPIDD^2 controller.

5.3. Design of IRUN algorithm based RPIDD^2 controller via master–slave model

In this study, the following description of the related control problem's mathematical model is adopted to describe the system under consideration as a minimization problem such that the controller parameters can be optimized. To find a feasible and efficient solution, the problem is represented as follows:

$$\boldsymbol{X} = [K_p,\ K_i,\ K_d,\ K_{dd},\ N_1, N_2]^T \tag{31}$$

The standard form of optimal control typically involves minimizing or maximizing a certain objective while satisfying system dynamics and constraints. In this context, the ISE objective function, given in Eq. (32), is used alongside the IRUN algorithm for minimization in order to handle this problem [45]:

$$ISE(X) = \int_0^{\infty} [e^*(t)]^2 dt \tag{32}$$

where $e^*(t)$ denotes the error between the outputs of the ideal reference model and the cruise control system.

The ISE objective function is a mathematical representation used in control theory to quantify the performance of a control system. It is commonly employed in various fields, including engineering, economics, and physics, to design and analyze systems that require optimal control. In the context of Eq. (32), the integration captures the cumulative effect of the control error over time. The "squared error" represents the difference between the desired or reference value (setpoint) and the actual value (output) of the controlled system. By squaring this difference, it is ensured that both positive and negative errors contribute positively to the cost, while also penalizing larger errors more significantly.

The above-mentioned design is subjected to the constraints of $OS \leq 0.5\%$ and $T_s \leq 1s$ as the design requirements of the time domains. In here, OS and T_s denote the percent overshoot and settling time, respectively. Besides variable ranges for the controller parameters are also used as $1 \leq K_p, K_d \leq 10$; $0.1 \leq K_i, K_{dd} \leq 1$ and $100 \leq N_1, N_2 \leq 1000$. These constraints ensure that the control actions remain within a feasible and safe range. Fig. 4 illustrates the application of the proposed master–slave approach to vehicle cruise control system.

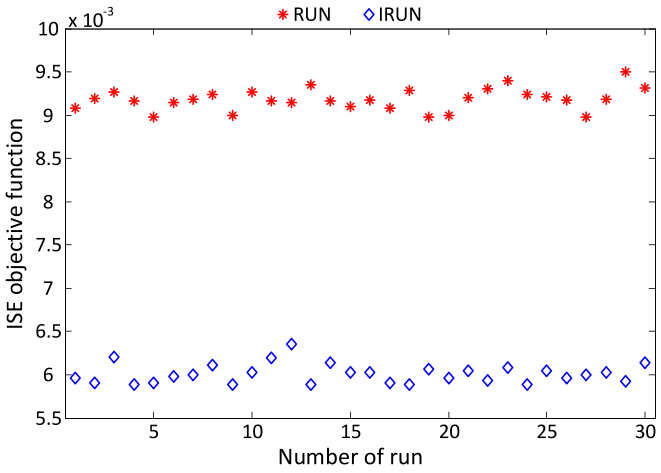


Fig. 5. Scatter plot for all 30 runs of the RUN and IRUN algorithms.

6. Comparative simulation results and discussion

6.1. Statistical performance

In this subsection, we present comparative statistical performance evaluation of the RUN and IRUN algorithms to highlight the superiority of the IRUN algorithm over the RUN algorithm. The specific experimental setup of 30 runs, 50 iterations, and a population size of 30 were considered to provide insights into the performance of both algorithms. Fig. 5 displays a scatter plot that represents the performance of the RUN and IRUN algorithms across all 30 runs. Each data point corresponds to a

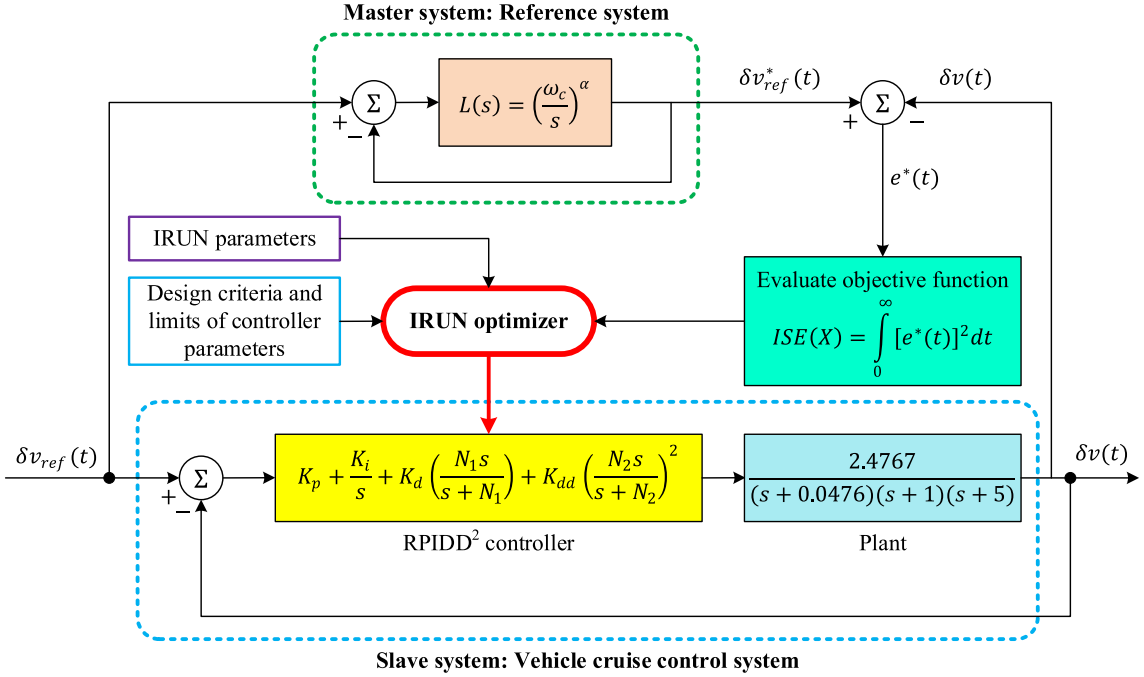


Fig. 4. Application of the proposed master–slave approach to vehicle cruise control system.

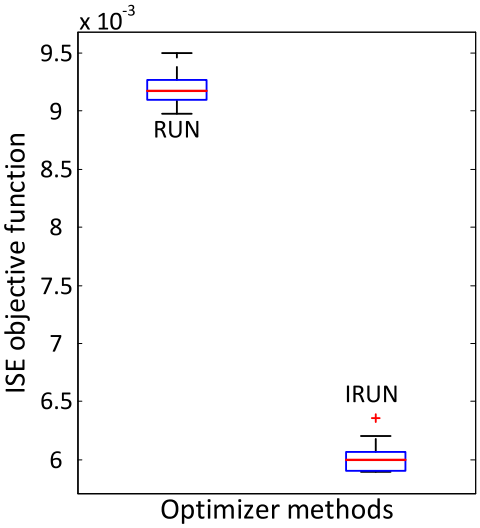


Fig. 6. Comparative boxplot analysis of the RUN and IRUN algorithms.

Table 2
Statistical and Wilcoxon signed-rank test results.

Metric	RUN	IRUN
Average	9.1815E − 03	6.0128E − 03
Standard Deviation	1.2825E − 04	1.1333E − 04
Minimum	8.9750E − 03	5.8907E − 03
Maximum	9.5046E − 03	6.3561E − 03
p-value	1.7344E − 06	

specific run. Observing the scatter plot, it is evident that all of data points for the IRUN algorithm lie below those for the RUN algorithm. This spatial distribution reinforces the notion that the IRUN algorithm consistently outperforms RUN in terms of minimizing ISE objective function. The clear separation of the data points implies a significant improvement in performance achieved by the IRUN algorithm.

To further investigate the performance disparity, Fig. 6 presents a comparative boxplot analysis of the RUN and IRUN algorithms. The boxplots provide a visual representation of the distribution and statistical summary of the metric values obtained by each algorithm. The boxplot for IRUN exhibits a lower median value and a narrow interquartile range compared to the boxplot for RUN. This finding suggests that the IRUN algorithm consistently achieves better results, with less variation, across the 30 runs.

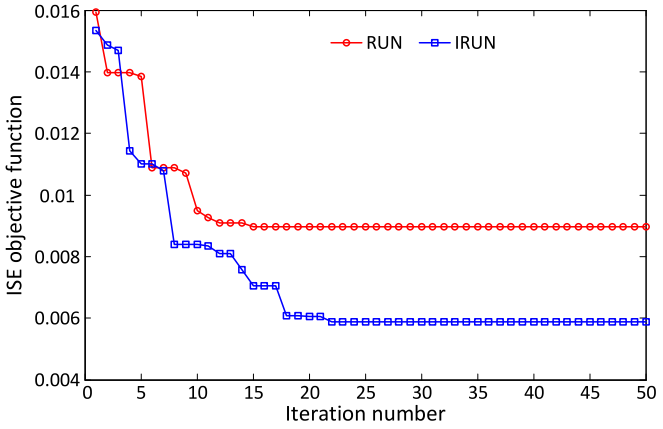


Fig. 7. Convergence profile of objective function for RUN and IRUN algorithms.

IRUN algorithm demonstrates clear superiority over the RUN algorithm across all measures. The average value of the metric achieved by the IRUN algorithm is significantly lower than that of RUN. This finding suggests that, on average, IRUN outperforms RUN in terms of the evaluated metric. Furthermore, the standard deviation of IRUN is lower than that of RUN, indicating that the results produced by IRUN are more consistent and less variable compared to RUN. A lower standard deviation suggests improved reliability and stability of the IRUN algorithm. The minimum and maximum values for IRUN are both lower than the corresponding values for RUN. This finding implies that even in the best and worst cases, IRUN performs better than RUN, further highlighting its superiority. Additionally, the Wilcoxon signed-rank test yielded a highly significant p-value of 1.7344E-06, providing strong evidence for a statistically significant difference between the two algorithms. This result supports the conclusion that the IRUN algorithm is superior to the RUN algorithm based on the evaluated metric.

6.2. Derived controller parameters and convergence curves

To be able to conduct analysis, we first obtained the controller parameters via RUN and IRUN algorithms. The best controller parameters obtained via RUN optimizer are $K_p = 5.0501$, $K_i = 0.1696$, $K_d = 5.5095$, $K_{dd} = 0.2369$, $N_1 = 380.0632$ and $N_2 = 116.6292$. Using those values and the transfer functions in the feedback loop provided in Fig. 3, the following closed loop transfer function is obtained for RUN optimizer.

$$CLTF_{RUN}(s) = \frac{1.318e04s^4 + 4.251e06s^3 + 7.182e07s^2 + 6.47e07s + 2.172e06}{s^7 + 619.4s^6 + 1.06e05s^5 + 5.805e06s^4 + 3.606e07s^3 + 9.917e07s^2 + 6.593e07s + 2.172e06} \tag{33}$$

Table 2, on the other hand, summarizes the statistical results obtained from the experiments, and the Wilcoxon signed-rank test was performed to determine the significance of the differences between the algorithms. This table presents the key statistical measures, including average, standard deviation, minimum, and maximum values for the metric along with the non-parametric statistical metric. Notably, the

Similarly, the best controller parameters obtained via IRUN algorithm are $K_p = 6.1660$, $K_i = 0.1972$, $K_d = 6.8975$, $K_{dd} = 0.5417$, $N_1 = 970.9724$ and $N_2 = 321.6791$. Using those values and the transfer functions in the feedback loop provided in Fig. 3, the following closed loop transfer function is obtained for IRUN algorithm.

$$CLTF_{IRUN}(s) = \frac{1.554e05s^4 + 1.455e08s^3 + 1.727e09s^2 + 1.535e09s + 4.907e07}{s^7 + 1620s^6 + 7.379e05s^5 + 1.05e08s^4 + 7.57e08s^3 + 2.259e09s^2 + 1.559e09s + 4.907e07} \tag{34}$$

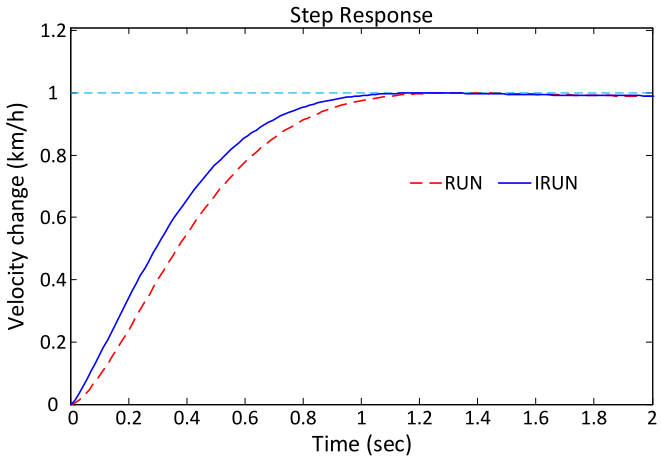


Fig. 8. Speed change related step response of the cruise control system with RUN and IRUN algorithms.

Table 3
PID parameters obtained via different available approaches and their corresponding closed loop transfer functions.

Ref.	Tuning method	Parameters and transfer function	Value
[22]	ALO-PID	K_p, K_i, K_d	3.858, 0.2089, 3.8581
		$CLTF(s)$	$\frac{9.555s^2+9.555s+0.5174}{s^4+6.048s^3+14.84s^2+9.793s+0.5174}$
[23]	RSALF-PID	K_p, K_i, K_d	4.3714, 0.1972, 4.7421
		$CLTF(s)$	$\frac{11.74s^2+10.83s+0.4884}{s^4+6.048s^3+17.03s^2+11.06s+0.4884}$
[11]	AOA-NM-PID	K_p, K_i, K_d	3.9840, 0.1798, 3.9976
		$CLTF(s)$	$\frac{9.901s^2+9.867s+0.4453}{s^4+6.048s^3+15.19s^2+10.11s+0.4453}$

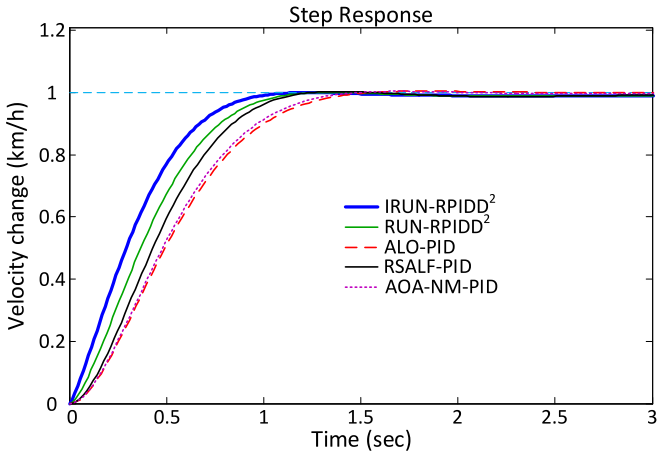


Fig. 9. Comparison of velocity step response of related system with different optimized PID controllers.

We performed the convergence profile assessment for RUN and IRUN algorithms in terms of minimizing the ISE objective function. Fig. 7 illustrates the convergence curves obtained via RUN and IRUN algorithms. As displayed in this figure, the proposed IRUN algorithm can reach lower objective function values in later iterations and can achieve the lowest value with respect to RUN optimizer.

6.3. Time domain stability evaluation

Time domain stability analysis was performed by considering the speed change step responses as demonstrated in Fig. 8. As can be observed, the proposed IRUN algorithm is able to reach smoother response with no overshoot, less rise and settling times making it the most convenient approach compared to RUN optimizer.

6.4. Comparisons with recent PID controllers

The selection of appropriate approaches in developing the proposed method holds paramount importance, as it determines the efficacy and superiority of the results obtained. To showcase the promise of our approach, rigorous comparisons were conducted against PID controllers tuned using various existing methodologies found in the literature. Specifically, we evaluated our approach against well-performing techniques, including the antlion optimizer-tuned PID (ALO-PID) controller [22], the enhanced reptile search algorithm with Lévy flight-tuned PID controller (RSALF-PID) [23], and the improved arithmetic optimization algorithm-tuned PID controller (AOA-NM-PID) [11]. By utilizing these established approaches, we were able to critically assess and validate the performance of our proposed method. The corresponding gain parameters obtained by the compared approaches, along with their corresponding closed-loop transfer functions, are presented in Table 3. These comparisons serve as a solid foundation for highlighting the uniqueness and superiority of our approach in achieving enhanced control performance and stability.

The true significance of the proposed IRUN approach becomes evident when comparing its performance to other established methods. This is evident in the comparative step responses presented in Fig. 9, where the effectiveness of the IRUN-RPIDD^2 approach is clearly demonstrated. The figure showcases the superior transient response capability of our proposed approach when it comes to accommodating velocity changes in the cruise control system. The distinctive and remarkable performance of our approach is visually evident, solidifying its prominence in achieving exceptional control performance.

Furthermore, an in-depth analysis of various time-domain performance metrics was conducted, aiming to gain further insights into the behavior of the proposed method. As depicted in Fig. 10, the proposed approach outperforms alternative optimized PID controllers in terms of crucial time-domain metrics such as percent overshoot, rise time, settling time, and peak time. The clear superiority of our approach in these performance measures reaffirms its effectiveness and confirms its potential to elevate the overall control capabilities of the cruise control system. These findings underscore the significance of the proposed IRUN approach, demonstrating its ability to surpass existing methods in terms of both transient response and time-domain performance metrics. The results obtained serve as a strong testament to the efficacy and potential impact of our approach in enhancing the control performance and overall efficiency of the cruise control system.

6.5. Comparisons with recent FOPID controllers

In order to thoroughly assess the effectiveness of our study, a comprehensive investigation was conducted by comparing the proposed approach with FOPID controllers tuned using diverse methodologies found in the literature. This rigorous analysis allowed us to evaluate the performance of our approach against well-established techniques, including the Harris hawks optimization-tuned FOPID (HHO-FOPID) controller [24], the artificial hummingbird algorithm-tuned FOPID (AHA-FOPID) controller [25], and the modified artificial hummingbird algorithm-tuned FOPID (m-AHA-FOPID) controller [25]. These selected approaches have been widely recognized for their efficacy and performance, making them suitable candidates for comparison. The gain parameters obtained by the compared approaches, along with their corresponding closed-loop transfer functions, are provided in Table 4.

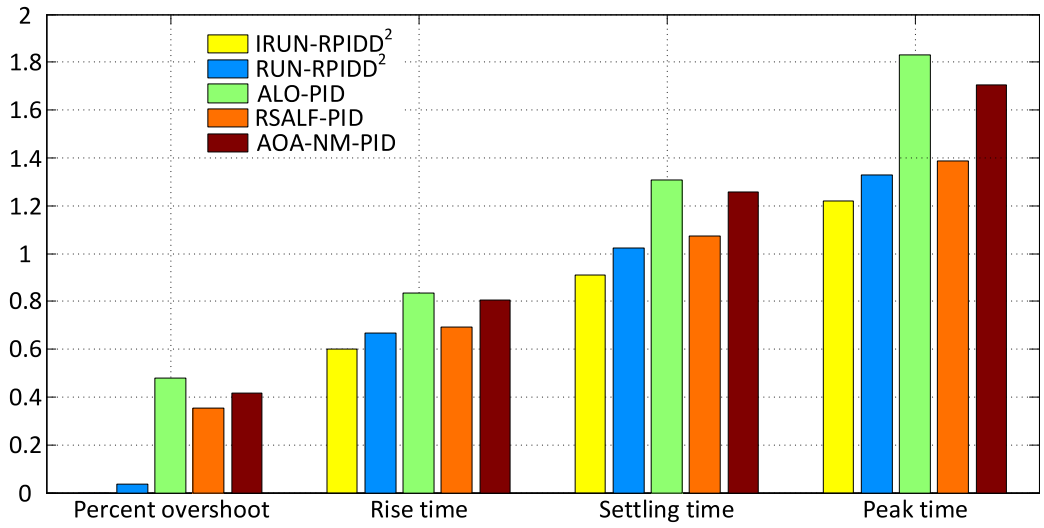


Fig. 10. Visual comparison of time-domain performance metrics with different optimized PID controllers.

Table 4
FOPID parameters obtained via different available approaches and their corresponding closed loop transfer functions.

Ref.	Tuning method	Parameters and transfer function	Value
[24]	HHO-FOPID	$K_p, K_i, K_d, \lambda, \mu$ $CLTF(s)$	4.4816, 0.2261, 4.6939, 0.9523, 1.0217 $\frac{11.625s^{1.974} + 11.099s^{0.9523} + 0.55997}{s^{3.9523} + 6.0476s^{2.9523} + 11.625s^{1.974} + 5.2856s^{1.9523} + 11.337s^{0.9523} + 0.55997}$
[25]	AHA-FOPID	$K_p, K_i, K_d, \lambda, \mu$ $CLTF(s)$	4.7183, 0.2127, 4.8352, 0.9528, 1.0280 $\frac{11.975s^{1.9808} + 11.686s^{0.9528} + 0.52679}{s^{3.9528} + 6.0476s^{2.9528} + 11.975s^{1.9808} + 5.2856s^{1.9528} + 11.924s^{0.9528} + 0.52679}$
[25]	m-AHA-FOPID	$K_p, K_i, K_d, \lambda, \mu$ $CLTF(s)$	4.9115, 0.1754, 4.9849, 0.9603, 1.0425 $\frac{12.346s^{2.0028} + 12.164s^{0.9603} + 0.43441}{s^{3.9603} + 6.0476s^{2.9603} + 12.346s^{2.0028} + 5.2856s^{1.9603} + 12.402s^{0.9603} + 0.43441}$

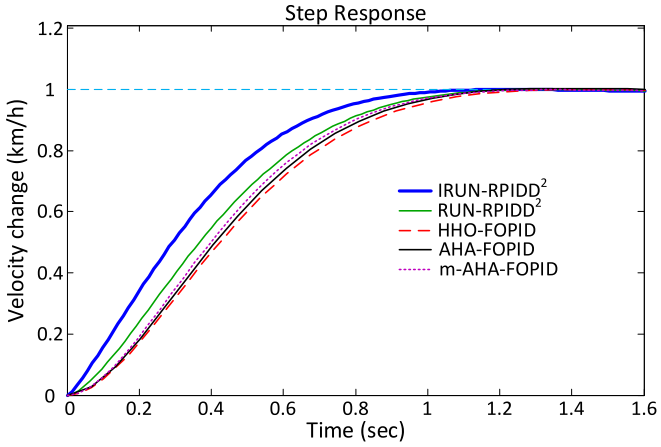


Fig. 11. Comparison of velocity step response of related system with different optimized FOPID controllers.

This meticulous comparison serves as a robust foundation for highlighting the significance of our proposed approach. By benchmarking against these well-performing methodologies, we were able to assess the superiority of our approach in terms of control performance and stability. The results obtained provide compelling evidence of the effectiveness and potential impact of our proposed approach in enhancing the control capabilities of the vehicle cruise control system.

The significance of the proposed approach becomes evident when comparing its performance to other existing methods. This is clearly demonstrated through the comparative step responses presented in Fig. 11, where the superior transient response capability of the proposed IRUN-RPIDD^2 approach for velocity changes in the cruise control system is prominently displayed. The figure unmistakably showcases the remarkable performance achieved by our approach, underlining its significance in achieving exceptional control capabilities.

Furthermore, an in-depth analysis of various time-domain performance metrics was conducted to gain further insights into the behavior of the proposed method. As depicted in Fig. 12, the proposed approach outperforms alternative optimized FOPID controllers across crucial time-domain performance measures, including percent overshoot, rise time, settling time, and peak time. These results firmly establish the superiority of our approach in achieving enhanced control performance and stability. The observed improvements in both transient response capability and time-domain performance metrics underscore the significance of the proposed approach. Its superior performance serves as a testament to its potential impact on enhancing the overall control accuracy and efficiency of the cruise control system. The findings further validate the effectiveness of our approach, positioning it as a noteworthy advancement in the field of vehicle cruise control systems.

6.6. *Comparisons with recent RPIDD^2 controllers*

Finally, to fully substantiate the promise of our proposed approach, comprehensive comparisons were conducted with RPIDD^2 controllers tuned using various approaches documented in the literature. This thorough analysis involved benchmarking against the performance of particle swarm optimization algorithm-tuned RPIDD^2 (PSO-RPIDD^2) controller [17], salp swarm algorithm-tuned RPIDD^2 (SSA-RPIDD^2) controller [17] and modified oppositional chaotic local search strategy-based aquila optimizer-tuned RPIDD^2 (CmOBL-AO-RPIDD^2) controller [17]. These selected approaches, known for their effectiveness and

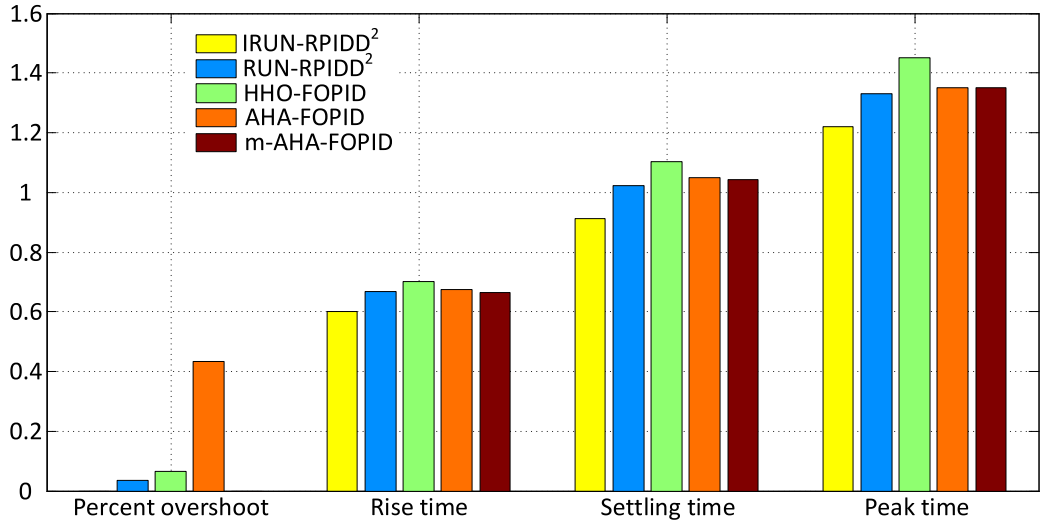


Fig. 12. Visual comparison of time-domain performance metrics with different optimized FOPID controllers.

Table 5
RPIDD^2 parameters obtained via different available approaches and their corresponding closed loop transfer functions.

Ref.	Tuning method	Parameters and transfer function	Value
[17]	PSO-RPIDD^2	$K_p, K_i, K_d, K_{dd}, N_1, N_2$ $CLTF(s)$	4.2666, 0.2860, 4.4874, 0.1022, 925.8360, 244.2857 $\frac{2.54e04s^4 + 1.903e07s^3 + 6.194e08s^2 + 5.842e08s + 3.913e07}{s^7 + 1420s^6 + 5.206e05s^5 + 5.838e07s^4 + 3.559e08s^3 + 9.116e08s^2 + 5.973e08s + 3.913e07}$
[17]	SSA-RPIDD^2	$K_p, K_i, K_d, K_{dd}, N_1, N_2$ $CLTF(s)$	4.9192, 0.2443, 5.1814, 0.2888, 681.5607, 564.9932 $\frac{2.371e05s^4 + 1.655e08s^3 + 2.805e09s^2 + 2.651e09s + 1.316e08}{s^7 + 1818s^6 + 1.1e06s^5 + 2.244e08s^4 + 1.487e09s^3 + 3.955e09s^2 + 2.703e09s + 1.316e08}$
[17]	CmOBL-AO-RPIDD^2	$K_p, K_i, K_d, K_{dd}, N_1, N_2$ $CLTF(s)$	5.1645, 0.4087, 5.9490, 0.3123, 764.4678, 389.4943 $\frac{1.286e05s^4 + 9.85e07s^3 + 1.718e09s^2 + 1.484e09s + 1.174e08}{s^7 + 1550s^6 + 7.566e05s^5 + 1.206e08s^4 + 8.038e08s^3 + 2.331e09s^2 + 1.512e09s + 1.174e08}$

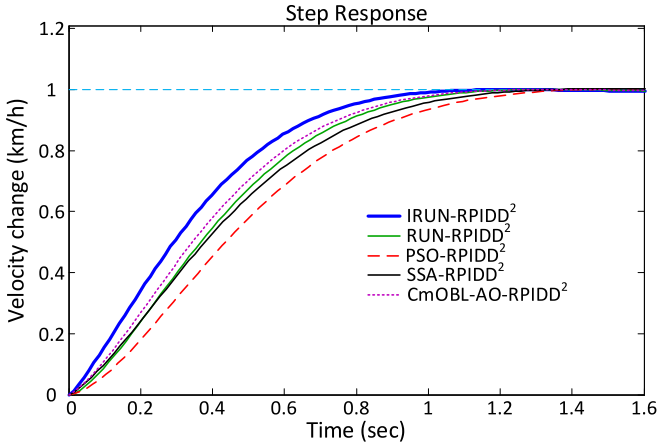


Fig. 13. Comparison of velocity step response of related system with different optimized RPIDD^2 controllers.

performance, provided a solid basis for evaluating the capabilities of our proposed approach. The gain parameters obtained by the compared approaches, alongside their corresponding closed-loop transfer functions, are meticulously presented in Table 5. By undertaking these rigorous comparisons, we aimed to showcase the superiority and effectiveness of our proposed approach in enhancing the performance of RPIDD^2 controllers. The results obtained from these comparisons provide compelling evidence of the advancements achieved by our proposed approach. Through this comprehensive evaluation, we highlight the significant contributions of our approach in terms of control accuracy, stability, and overall system performance. The findings underscore the potential impact of our approach in the field of vehicle cruise control systems, positioning it as a valuable addition to the existing literature.

The superiority of the proposed IRUN-RPIDD^2 approach over alternative methods is visually demonstrated through comparative step responses, as depicted in Fig. 13. It is evident from the figure that our proposed approach exhibits significantly improved transient response capability for velocity changes in the cruise control system. The clear distinction in performance further emphasizes the effectiveness of our approach in achieving superior control capabilities. To gain a comprehensive understanding of the proposed method's behavior, an analysis of time-domain performance metrics was conducted for various optimized RPIDD^2 controllers. Fig. 14 illustrates the outcomes of this analysis, showcasing the superiority of our proposed approach across different time-domain performance metrics, including percent overshoot, rise time, settling time, and peak time. These results provide empirical evidence of the remarkable performance achieved by our approach in enhancing the overall control performance and stability of the cruise control system. The notable improvements in both transient response capability and time-domain performance metrics reaffirm the significance of the proposed approach. Its exceptional performance positions it as a valuable contribution to the field of vehicle cruise control systems. The findings highlight the potential impact of our approach in advancing control accuracy, efficiency, and overall system performance.

6.7. Robustness validation of the proposed method

In order to assess the robustness of the proposed IRUN based design approach, a comprehensive analysis was conducted under varying parameter scenarios. The time-domain performance metrics were

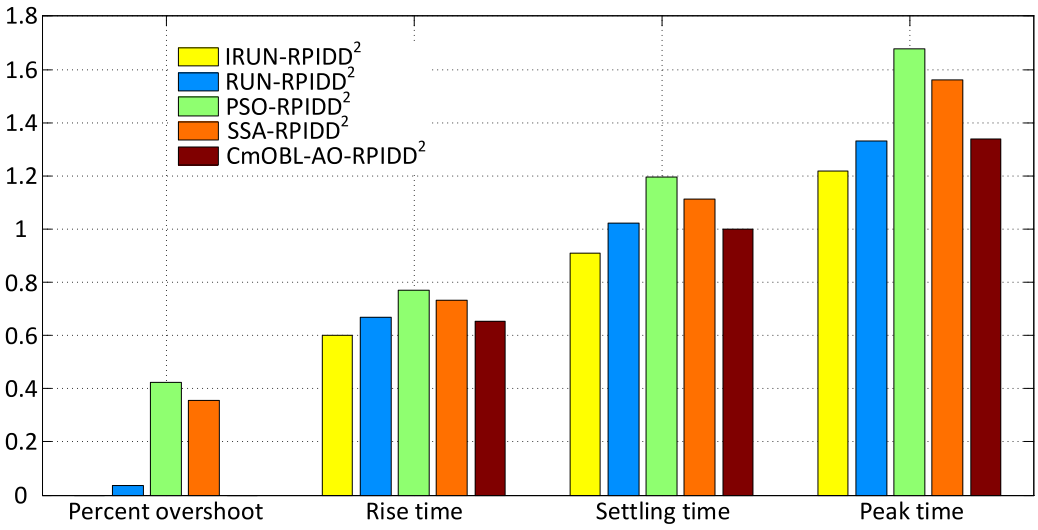


Fig. 14. Visual comparison of time-domain performance metrics with different optimized RPIDD^2 controllers.

Table 6
Comparison of time-domain performance metrics under parameter changes.

Scenario	Algorithm	Overshoot (%)	Rise time (sec)	Settling time (sec)	Peak time (sec)
Nominal	RUN	0.0359	0.6681	1.0229	1.3298
($M = 1500, \tau = 0.2$)	IRUN	**0**	**0.6023**	**0.9106**	**1.2197**
Scenario A	RUN	3.2339	0.5970	1.4237	1.1752
($M = 1350, \tau = 0.24$)	IRUN	**2.7197**	**0.5405**	**1.2426**	**1.0618**
Scenario B	RUN	2.0161	0.7015	1.4125	1.3872
($M = 1600, \tau = 0.25$)	IRUN	**1.9776**	**0.6295**	**0.8939**	**1.2374**
Scenario C	RUN	0.0333	0.6254	0.9563	1.2540
($M = 1400, \tau = 0.19$)	IRUN	**0**	**0.5666**	**0.8615**	**1.1545**
Scenario D	RUN	**0**	0.7031	1.1612	8.6753
($M = 1550, \tau = 0.18$)	IRUN	**0**	**0.6327**	**1.0315**	**1.4277**

compared between the IRUN and traditional RUN algorithms across different scenarios. The results from this analysis provide valuable insights into the robustness of the IRUN approach in maintaining desirable performance characteristics even under parameter changes. Four distinct scenarios, labeled as A, B, C, and D, were considered to evaluate the robustness of the proposed method. Each scenario involves changes in the M and τ parameters, thereby simulating different operating conditions. The nominal scenario, represented by $M = 1500$ and $\tau = 0.2$, served as the baseline for comparison. Scenarios A, B, C, and D encompass variations in either or both of these parameters as summarized in Table 6.

The time-domain performance metrics analyzed for comparison included overshoot, rise time, settling time, and peak time. These metrics provide insights into the transient and steady-state behavior of the system under different parameter scenarios. The performance of the IRUN algorithm was evaluated alongside the traditional RUN algorithm for each scenario, allowing a direct comparison of their robustness. The robustness of the IRUN based design approach is evident from the comparison of time-domain performance metrics under parameter changes. Notably, in the nominal scenario, IRUN demonstrated an exceptional performance with zero overshoot, indicating its capability to eliminate oscillations in the system response. Additionally, the rise time, settling time, and peak time of IRUN were better than those of the traditional RUN algorithm, showcasing its effectiveness in maintaining system stability and transient response quality. Therefore, it can be observed that the proposed IRUN algorithm has a superiority over the RUN algorithm in terms of time-domain performance metrics in both the nominal condition and all potential scenarios. The maximum deviation for overshoot is 2.7197%, for rise time is 0.0618s, for settling time is 0.3320s, and for peak time is 0.2080s. Since the deviation values are at reasonable levels, it can be stated that the suggested IRUN-based control approach is minimally affected by changes in system parameters. The robustness analysis conducted in this section underscores the efficacy of the proposed IRUN based design approach. The IRUN algorithm consistently maintained favorable performance metrics across diverse parameter scenarios, showcasing its ability to deliver stable and efficient system responses.

7. Conclusion

This study has presented a comprehensive investigation into enhancing the time-domain performance and stability of vehicle cruise control systems. By introducing the IRUN optimizer and adopting a master–slave approach with the RPIDD^2 controller, significant advancements have been achieved in control accuracy and efficiency. The application of metaheuristic algorithms, as exemplified by the IRUN optimizer, has demonstrated remarkable potential in optimizing the control system. The incorporation of advanced strategies, such as quadratic interpolation, Laplacian segment mutation, Levy flight, and information-sharing-based local search mechanisms, has enhanced the optimization capabilities of the IRUN algorithm, making it highly effective for tuning the RPIDD^2 controller.

Through extensive experimentation and evaluation, the superiority and effectiveness of the proposed approach have been substantiated. Statistical performance analysis, Wilcoxon's test, convergence analyses, and time-domain-based stability evaluations have all confirmed the enhanced capabilities of the IRUN-based RPIDD^2 controller within the master–slave system. Furthermore, comprehensive comparative studies have been conducted, benchmarking the proposed approach against well-established methods utilizing PID, FOPID, and RPIDD^2 controllers.

The results have demonstrated the superior performance of the proposed approach, showcasing its ability to outperform existing optimization techniques across a range of scenarios. Moreover, the robustness analysis further validated the efficacy of the proposed method. The implications of this research are far-reaching, as it not only contributes to the advancement of vehicle cruise control systems but also addresses the broader issue of traffic safety. By improving control accuracy and efficiency, the proposed approach has the potential to reduce the occurrence of traffic accidents caused by human error. In summary, this study has provided a comprehensive framework for enhancing the time-domain performance and stability of vehicle cruise control systems. The integration of the IRUN optimizer and the master–slave approach with the RPIDD^2 controller has yielded significant improvements, paving the way for safer and more efficient driving experiences. The findings of this research offer valuable insights and open up new avenues for future advancements in the field of automotive engineering.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] B. Chen, D. Sun, J. Zhou, W. Wong, Z. Ding, A future intelligent traffic system with mixed autonomous vehicles and human-driven vehicles, Inf. Sci. (N Y). 529 (2020) 59–72, https://doi.org/10.1016/j.ins.2020.02.009.

[2] M. Hasenjager, M. Heckmann, H. Wersing, A survey of personalization for advanced driver assistance systems, IEEE Trans. Intell. Veh. 5 (2020) 335–344, https://doi.org/10.1109/TIV.2019.2955910.

[3] X. Gao, X. Bian, Autonomous driving of vehicles based on artificial intelligence, J. Intell. Fuzzy Syst. 41 (2021) 4955–4964, https://doi.org/10.3233/JIFS-189982.

[4] J. Huang, Y. Chen, X. Peng, L. Hu, D. Cao, Study on the driving style adaptive vehicle longitudinal control strategy, IEEE/CAA J. Autom. Sin. 7 (2020) 1107–1115, https://doi.org/10.1109/JAS.2020.1003261.

[5] S. Vasebi, Y.M. Hayeri, A.M. Saghiri, A literature review of energy optimal adaptive cruise control algorithms, IEEE Access 11 (2023) 13636–13646, https://doi.org/10.1109/ACCESS.2023.3241140.

[6] M. Liu, W. Chen, J. Huang, Y. Ning, Kalman filter-based model predictive control for an adaptive cruise control system considering measurement noise, SAE Int. J. Connected Automated Vehicles. 3 (2020) 12-03-01–0005. https://doi.org/10.4271/12-03-01-0005.

[7] D. Theodosis, I. Karafyllis, M. Papageorgiou, Cruise controllers for lane-free ring-roads based on control Lyapunov functions, J. Franklin Inst. 360 (2023) 6131–6161, https://doi.org/10.1016/j.jfranklin.2023.02.003.

[8] S. Ekinci, D. Izci, L. Abualigah, A.G. Hussien, C.-L. Thanh, S. Khatir, Revolutionizing vehicle cruise control: an elite opposition-based pattern search mechanism augmented INFO algorithm for enhanced controller design, Int. J. Comput. Intelligence Syst. 16 (2023) 129, https://doi.org/10.1007/s44196-023-00304-8.

[9] S. Chaturvedi, N. Kumar, Design and implementation of an optimized PID controller for the adaptive cruise control system, IETE J. Res. (2021) 1–8, https://doi.org/10.1080/03772063.2021.2012282.

[10] D. Izci, S. Ekinci, A novel hybrid ASO-NM algorithm and its application to automobile cruise control system, in: G. Mathur, M. Bundele, L. Mahendra, M. Paprzycki (Eds.), 2nd International Conference on Artificial Intelligence: Advances and Applications, 1st ed., Springer Singapore, 2022: pp. 333–343. https://doi.org/10.1007/978-981-16-6332-1_29.

[11] D. Izci, S. Ekinci, M. Kayri, E. Eker, A novel improved arithmetic optimization algorithm for optimal design of PID controlled and Bode's ideal transfer function based automobile cruise control system, Evol. Syst. 13 (2022) 453–468, https://doi.org/10.1007/s12530-021-09402-4.

[12] R. Pradhan, B.B. Pati, Optimal FOPID Controller for an Automobile Cruise Control System, in: 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE), IEEE, 2018: pp. 1436–1440. https://doi.org/10.1109/ICRIEECE44171.2018.9008957.

[13] R. Zhang, N. Wu, Z. Wang, K. Li, Z. Song, Z. Chang, X. Chen, F. Yu, Constrained hybrid optimal model predictive control for intelligent electric vehicle adaptive cruise using energy storage management strategy, J. Energy Storage. 65 (2023), 107383, https://doi.org/10.1016/j.est.2023.107383.

[14] E. Landolfi, A. Salvi, A. Troiano, C. Natale, Hardware-in-the-loop validation of an adaptive model predictive control on a connected and automated vehicle, Int. J. Adapt Control Signal Process. 37 (2023) 1459–1491, https://doi.org/10.1002/acs.3583.

[15] D.C. Selvaraj, S. Hegde, N. Amati, F. Deflorio, C.F. Chiasserini, A deep reinforcement learning approach for efficient, safe and comfortable driving, Appl. Sci. 13 (2023) 5272, https://doi.org/10.3390/app13095272.

[16] M. Zhu, G. Tan, Research on Cooperative Adaptive Cruise Control (CACC) Based on Fuzzy PID Algorithm, in: 2023. https://doi.org/10.4271/2023-01-0682.

[17] S. Ekinci, D. Izci, L. Abualigah, R.A. Zitar, A modified oppositional chaotic local search strategy based aquila optimizer to design an effective controller for vehicle cruise control system, J. Bionic Eng. 20 (2023) 1828–1851, https://doi.org/10.1007/s42235-023-00336-y.

[18] D. Izci, S. Ekinci, An improved RUN optimizer based real PID plus second-order derivative controller design as a novel method to enhance transient response and robustness of an automatic voltage regulator, E-Prime – Adv. Electr. Eng., Electron. Energy 2 (2022), 100071, https://doi.org/10.1016/j.prime.2022.100071.

[19] I. Ahmadianfar, A.A. Heidari, A.H. Gandomi, X. Chu, H. Chen, RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method, Expert Syst. Appl. 181 (2021), 115079, https://doi.org/10.1016/j.eswa.2021.115079.

[20] S. Ekinci, D. Izci, M. Kayri, An effective controller design approach for magnetic levitation system using novel improved manta ray foraging optimization, Arab. J. Sci. Eng. 47 (2022) 9673–9694, https://doi.org/10.1007/s13369-021-06321-z.

[21] D. Izci, S. Ekinci, S. Mirjalili, L. Abualigah, An intelligent tuning scheme with a master/slave approach for efficient control of the automatic voltage regulator, Neural Comput. & Applic. 35 (2023) 19099–19115, https://doi.org/10.1007/s00521-023-08740-5.

[22] R. Pradhan, S.K. Majhi, J.K. Pradhan, B.B. Pati, Antlion optimizer tuned PID controller based on Bode ideal transfer function for automobile cruise control system, J. Ind. Inf. Integr. 9 (2018) 45–52, https://doi.org/10.1016/j.jii.2018.01.002.

[23] S. Ekinci, D. Izci, Enhanced reptile search algorithm with Lévy flight for vehicle cruise control system design, Evol. Intel. 16 (2023) 1339–1351, https://doi.org/10.1007/s12065-022-00745-8.

[24] D. Izci, S. Ekinci, An efficient FOPID controller design for vehicle cruise control system using HHO algorithm, in: 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), IEEE, 2021: pp. 1–5. https://doi.org/10.1109/HORA52670.2021.9461336.

[25] L. Abualigah, S. Ekinci, D. Izci, R. Abu Zitar, Modified elite opposition-based artificial hummingbird algorithm for designing FOPID controlled cruise control system, Intelligent Automation & Soft Computing. (2023). https://doi.org/10.32604/iasc.2023.040291.

[26] D. Izci, S. Ekinci, S. Mirjalili, Optimal PID plus second-order derivative controller design for AVR system using a modified Runge Kutta optimizer and Bode's ideal reference model, Int. J. Dyn. Control. 11 (2023) 1247–1264, https://doi.org/10.1007/s40435-022-01046-9.

[27] R.M. Rizk-Allah, A.E. Hassanien, A comprehensive survey on the sine–cosine optimization algorithm, Artif. Intell. Rev. 56 (2023) 4801–4858, https://doi.org/10.1007/s10462-022-10277-3.

[28] W. Guo, Y. Wang, F. Dai, P. Xu, Improved sine cosine algorithm combined with optimal neighborhood and quadratic interpolation strategy, Eng. Appl. Artif. Intel. 94 (2020), 103779, https://doi.org/10.1016/j.engappai.2020.103779.

[29] X. Weng, A.A. Heidari, G. Liang, H. Chen, X. Ma, M. Mafarja, H. Turabieh, Laplacian Nelder-Mead spherical evolution for parameter estimation of photovoltaic models, Energy Convers. Manag. 243 (2021), 114223, https://doi.org/10.1016/j.enconman.2021.114223.

[30] R.M. Rizk-Allah, A.E. Hassanien, D. Song, Chaos-opposition-enhanced slime mould algorithm for minimizing the cost of energy for the wind turbines on high-altitude sites, ISA Trans. 121 (2022) 191–205, https://doi.org/10.1016/j.isatra.2021.04.011.

[31] O. Tarkhaneh, H. Shen, Training of feedforward neural networks for data classification using hybrid particle swarm optimization, Mantegna Lévy flight and neighborhood search, Heliyon 5 (2019) e01275.

[32] S. Ekinci, D. Izci, M.R. Al Nasar, R. Abu Zitar, L. Abualigah, Logarithmic spiral search based arithmetic optimization algorithm with selective mechanism and its application to functional electrical stimulation system control, Soft. Comput. 26 (2022) 12257–12269, https://doi.org/10.1007/s00500-022-07068-x.

[33] D. Izci, 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 (2021) 3195–3211, https://doi.org/10.1177/01423312211019633.

[34] S. Das, I. Pan, S. Das, A. Gupta, Improved model reduction and tuning of fractional-order PIλDμ controllers for analytical rule extraction with genetic programming, ISA Trans. 51 (2012) 237–261, https://doi.org/10.1016/j.isatra.2011.10.004.

[35] D. Izci, S. Ekinci, H.L. Zeynelgil, J. Hedley, Performance evaluation of a novel improved slime mould algorithm for direct current motor and automatic voltage regulator systems, Trans. Inst. Meas. Control 44 (2022) 435–456, https://doi.org/10.1177/01423312211037967.

[36] M. Kumar, Y.V. Hote, Real-time performance analysis of PIDD2 controller for nonlinear twin rotor TITO aerodynamical system, J. Intell. Rob. Syst. 101 (2021) 55, https://doi.org/10.1007/s10846-021-01322-4.

[37] H.W. Bode, Network analysis and feedback amplifier design, Van Nostrand, 1945.

[38] R. Azarmi, M. Tavakoli-Kakhki, A.K. Sedigh, A. Fatehi, Robust fractional order PI controller tuning based on bode's ideal transfer function, IFAC-PapersOnLine. 49 (2016) 158–163, https://doi.org/10.1016/j.ifacol.2016.07.519.

[39] X. Li, Y. Wang, N. Li, M. Han, Y. Tang, F. Liu, Optimal fractional order PID controller design for automatic voltage regulator system based on reference model using particle swarm optimization, Int. J. Mach. Learn. Cybern. 8 (2017) 1595–1605, https://doi.org/10.1007/s13042-016-0530-2.

[40] C. Boussalem, R. Mansouri, M. Bettayeb, M. Hamerlain, Fractional Order Integral Controller Design Based on a Bode's Ideal Transfer Function: Application to the

Control of a Single Tank Process, in: 2021: pp. 155–169. https://doi.org/10.1007/978-981-15-6403-1_11.
[41] L. Zhang, Q. Zhang, W. Wang, Application of Ideal Bode Transfer Function Tuning Fractional Order PID in Pressure Difference of Vertical Mill, in: 2020 Chinese Control And Decision Conference (CCDC), IEEE, 2020: pp. 3501–3505. https://doi.org/10.1109/CCDC49329.2020.9163920.
[42] M.L. Frikh, F. Soltani, N. Bensiali, N. Boutasseta, N. Fergani, Fractional order PID controller design for wind turbine systems using analytical and computational tuning approaches, Comput. Electr. Eng. 95 (2021), 107410, https://doi.org/10.1016/j.compeleceng.2021.107410.
[43] E. Yumuk, M. Güzelkaya, İ. Eksin, A robust fractional-order controller design with gain and phase margin specifications based on delayed Bode's ideal transfer function, J. Franklin Inst. 359 (2022) 5341–5353, https://doi.org/10.1016/j.jfranklin.2022.05.033.
[44] D. Izci, S. Ekinci, B. Hekimoğlu, A novel modified Lévy flight distribution algorithm to tune proportional, integral, derivative and acceleration controller on buck converter system, Trans. Inst. Meas. Control 44 (2022) 393–409, https://doi.org/10.1177/01423312211036591.
[45] D. Izci, S. Ekinci, A novel improved version of hunger games search algorithm for function optimization and efficient controller design of buck converter system, E-Prime – Adv. Electr. Eng., Electron. Energy 2 (2022), 100039, https://doi.org/10.1016/j.prime.2022.100039.