applied sciences

MDPI

Article

A Differential Evolution-Enhanced Computational Optimization Framework for PID Tuning of Vehicle Cruise Control Systems Under Disturbance and Uncertainty

Cebrail Turkeri [1,2], Serdar Ekinci [3], Davut Izci [4,*] and Erdal Akin [3,5,*]

[1] School of Engineering, University of Warwick, Coventry CV4 7AL, UK; cebrail.turkeri.2@warwick.ac.uk
[2] Department of Computer Engineering, Batman University, Batman 72100, Türkiye
[3] Department of Computer Engineering, Bitlis Eren University, Bitlis 13100, Türkiye; sekinci@beu.edu.tr
[4] Department of Electrical and Electronic Engineering, Bursa Uludag University, Bursa 16059, Türkiye
[5] Department of Computer Science and Media Technology, Malmö University, 205 06 Malmö, Sweden
* Correspondence: davutizci@uludag.edu.tr (D.I.); erdal.akin@mau.se (E.A.)

Abstract

Simulation-based controller tuning for dynamic engineering systems requires computational optimization methods capable of maintaining search stability, solution consistency, and robust performance under uncertain operating conditions. In vehicle cruise control systems, accurate tuning of proportional–integral–derivative (PID) controllers remains challenging due to nonlinear dynamics, parameter variations, external disturbances, and changing operating conditions. Although metaheuristic optimization methods have been widely used for PID parameter tuning, their performance may be limited by an imbalanced exploration–exploitation trade-off, which can lead to premature convergence and inconsistent solution quality. To address these issues, this study proposes a hybrid centered collision optimizer with differential evolution, termed h-CCODE. The proposed framework incorporates differential evolution-based mutation and crossover mechanisms into the centered collision optimizer to enhance population diversity, convergence stability, and solution refinement while preserving the original collision-based search dynamics. The computational performance of h-CCODE is first evaluated using the CEC2022 benchmark suite, where the proposed method demonstrates improved convergence characteristics and competitive solution quality compared with the original optimizer and with four further metaheuristics. The method is then applied to simulation-based PID tuning of a vehicle cruise control system. Comparative results show that the h-CCODE-tuned PID controller provides a balanced transient response with reduced settling time and overshoot while maintaining a comparable rise time. Disturbance rejection, reference tracking, frequency-domain, and parameter uncertainty analyses further confirm the stable and consistent performance of the proposed control framework under non-ideal operating conditions. Overall, the results indicate that h-CCODE provides an effective computational optimization approach for simulation-based controller tuning in dynamic engineering systems.

Keywords: hybrid optimization; centered collision optimizer; differential evolution; PID controller; vehicle cruise control system

Check for updates

Received: 26 August 2026
Revised: 5 October 2026
Accepted: 9 October 2026
Published: 13 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Vehicle cruise control systems remain an important research topic in control engineering because they provide a practical framework for maintaining a desired vehicle

speed under varying driving conditions. From a dynamic perspective, the problem is inherently complex, as the longitudinal motion of the vehicle is influenced by aerodynamic drag, road slope, actuator dynamics, and external disturbances, all of which introduce nonlinearities and may degrade tracking performance if not properly addressed in the controller design [1,2]. The ongoing interest in this area is also reflected in recent studies on fractional-order control, adaptive cruise control under varying road conditions, cooperative cruise control, and intelligent vehicle systems with safety constraints, where improving regulation accuracy and robustness continues to be a central objective [3–6]. Despite the increasing availability of advanced control techniques, PID controllers remain widely used in practical applications due to their simple structure, ease of implementation, and transparent tuning characteristics [7]. However, the effectiveness of PID controllers is highly dependent on the proper selection of their parameters, and this tuning process becomes increasingly challenging in the presence of nonlinear behavior, parameter variations, and changing operating conditions [8].

This dependence on parameter selection has motivated a substantial body of work on advanced PID structures in which the gains are no longer treated as fixed quantities. In adaptive schemes, the controller parameters are adjusted online from measured signals, which allows the loop to follow variations in plant behavior, and recent formulations have combined such adaptation with fault tolerance and disturbance-observer mechanisms in order to preserve tracking accuracy under input saturation and multisource disturbances [9]. Fuzzy PID designs replace the fixed gain set with a rule base that maps the error and its rate of change onto gain corrections, an approach that has been shown to improve interaction accuracy in physically coupled systems where the disturbance is difficult to model explicitly [10]. Neural-network-based PID designs pursue a similar objective through learned mappings, in which the gains are produced by a network trained either offline or during operation. These approaches are effective when the plant is poorly identified or strongly time-varying, but the flexibility is obtained at a cost: online adaptation laws require a stability proof for each new configuration, fuzzy designs depend on a rule base and membership functions that must be devised for the specific application, and learning-based designs introduce a training stage together with a controller whose behavior is less transparent to certification. A fixed-gain PID controller whose parameters are determined offline avoids all three difficulties, since the implemented structure remains the one already accepted in industrial practice and no additional online mechanism is introduced into the loop. The quality of such a design, however, rests entirely on the tuning stage, which is precisely the stage addressed in the present study. It should further be noted that the contribution made here is located at the level of the tuning procedure rather than the controller structure, so the proposed optimizer is equally applicable to the parameter selection problems that arise within adaptive, fuzzy, and learning-based PID architectures, for instance in the calibration of scaling factors or adaptation gains.

Beyond the single vehicle, recent research has increasingly treated cruise control as part of cooperative multi-vehicle systems. In vehicle platoons, a systematic study of the factors governing performance has shown that network disconnections have the most severe effect on string stability, whereas external disturbances such as wind, road slope, and rolling resistance give rise to steady-state tracking errors, with vehicle heterogeneity, communication delays, and sensor or actuator faults also examined [11]. Within this setting, static output-feedback LQR control based on predecessor–leader following has been proposed to regulate inter-vehicle spacing while exploiting the aerodynamic drag reduction obtained in close-following formations to improve energy efficiency [12], and coordinated observer-based LQR schemes have been developed to shorten settling under disturbances [13]. These studies address the coordination layer of the platoon and synthesize the feedback law from a model of the whole

string of vehicles. The present work is concerned with a different level of the control hierarchy, namely the speed regulation of an individual vehicle, which in hierarchical architectures remains the inner loop through which the commands of an upper-level cruise or platoon controller are executed. Improvements in the tuning of this loop are therefore complementary to, rather than in competition with, the multi-vehicle strategies cited above and the same disturbance sources identified in [11].

To overcome these limitations, metaheuristic optimization-based tuning approaches have been widely investigated. Unlike classical tuning rules, such as the Ziegler–Nichols and Cohen–Coon methods, which rely on simplified assumptions and fixed operating conditions [14,15], metaheuristic algorithms provide a flexible framework for exploring the controller parameter space more effectively [16]. In the context of vehicle cruise control systems, various optimization techniques have been applied to PID controller design [17], including genetic algorithm (GA) [18], antlion optimizer (ALO) [19], arithmetic optimization-based methods (AOA) [20], reptile search-based approaches (RSA) [21], sine cosine algorithm (SCA) [22], and improved arithmetic optimization strategies [23]. In addition, more recent algorithms such as the slime mould algorithm (SMA) have also been employed, demonstrating competitive performance in terms of convergence and control accuracy [24]. Relative to these studies, the present work differs in two respects. The tuning is performed with an optimizer whose modification is motivated by a quantified deficiency of its base algorithm and whose benefit is verified statistically over repeated runs, and the resulting controller is assessed not only through its nominal step response but also in terms of disturbance and noise rejection, frequency-domain margins, reference tracking, and parametric uncertainty, the latter including a Monte Carlo analysis over randomly perturbed plants.

Although these studies confirm the effectiveness of optimization-based tuning, the achieved performance is often dependent on the search behavior of the selected algorithm. In particular, achieving a proper balance between exploration and exploitation remains a key challenge, as insufficient exploration may lead to premature convergence, while excessive exploration may slow down convergence and reduce solution quality [25–27]. These challenges become more pronounced in control applications, where the quality of the optimized parameters directly affects both transient and steady-state system performance.

In response to these limitations, hybrid optimization strategies have recently attracted considerable attention as a means of improving the performance of metaheuristic algorithms. The main idea behind hybridization is to combine complementary search mechanisms in order to enhance both diversification and intensification capabilities [28]. Among various approaches, the integration of differential evolution (DE) operators into existing optimization frameworks has been shown to improve convergence stability and solution quality in different engineering problems [29–33]. Differential evolution is particularly suitable for such integration due to its mutation and crossover mechanisms based on vector differences, which promote population diversity and facilitate effective exploration of the search space [34]. In addition, its greedy selection strategy contributes to stable convergence by preserving improved candidate solutions.

Despite these advantages, the potential of combining DE with recently developed optimization algorithms, particularly in control-oriented applications, remains relatively underexplored. In this regard, the centered collision optimizer (CCO), recently introduced as a physics-based metaheuristic algorithm, has demonstrated promising performance in solving complex optimization problems [35]. Inspired by collision dynamics, CCO employs a population-based search mechanism that enables effective exploration of the solution space. However, in common with many population-based methods, its search behavior is not equally reliable across landscape types [36,37]. The benchmark analysis

reported in Section 4 of this study shows that the difficulty is confined to rugged and multi-basin landscapes and manifests itself in two distinct forms: on several multimodal and hybrid functions the algorithm locates the global basin in some runs but stagnates elsewhere in others, producing a pronounced spread between the best and worst results over repeated executions, while on certain composition functions the entire population settles in an inferior basin in every run. Both forms are symptomatic of an exploration–exploitation imbalance rather than of an insufficient evaluation budget, and both concern the consistency of the search rather than its peak performance. Furthermore, the application of CCO to control system design, and in particular to PID controller tuning for vehicle cruise control systems, has not yet been sufficiently investigated.

Motivated by these observations, this study proposes a hybrid optimization approach by integrating differential evolution operators into the CCO framework, referred to as h-CCODE. The proposed method aims to improve convergence stability, enhance solution consistency, and strengthen the overall search capability of the original CCO while preserving its underlying collision-based dynamics. The effectiveness of the proposed approach is evaluated using both benchmark optimization problems and a vehicle cruise control system, allowing a comprehensive assessment of its performance in both numerical and practical control scenarios. The main contributions of this study can be summarized as follows:

- A diagnostic analysis of the centered collision optimizer is provided, in which two distinct failure modes of the baseline search are identified and quantified: a loss of run-to-run consistency on multimodal and hybrid landscapes, where the global basin is located but not retained, and a systematic confinement to an inferior basin on composition landscapes.
- A targeted hybrid, h-CCODE, is proposed in which the DE operators act as a weak refinement stage on the population after the decorrelated-space individuals have been mapped back, each operator being linked to a specific deficiency of the baseline. The modification removes both failure modes while leaving the behavior of CCO unchanged on the landscapes where it already converges to the optimum.
- To the best of the authors' knowledge, CCO and its DE-enhanced form are applied here to controller tuning for the first time. For the vehicle cruise control problem, h-CCODE achieves a statistically significant reduction in the tuning cost relative to CCO over 30 independent runs; obtains a lower mean tuning cost than particle swarm optimization, the grey wolf optimizer, the mountain gazelle optimizer, and quadratic interpolation optimization applied to the same problem under identical conditions; and yields a PID design with the shortest rise and settling times and the smallest overshoot among the compared designs.
- The tuned controller is characterized beyond its nominal step response, through disturbance and noise rejection, frequency-domain margins including the effect of the delay approximation, reference tracking, structured parameter variations, and a Monte Carlo analysis over 1000 perturbed plants, which together delimit the conditions under which the design remains valid.

The remainder of this paper is organized as follows. Section 2 presents the fundamentals of the centered collision optimizer, while its detailed mathematical formulation is provided in Appendix A. Section 3 introduces the proposed h-CCODE algorithm and explains the hybridization mechanism with differential evolution. Section 4 evaluates the optimization performance of the proposed method using benchmark test functions. Section 5 describes the vehicle cruise control system and formulates the control problem. Section 6 presents simulation results, comparative analyses, and robustness evaluations. Finally, Section 7 concludes the paper and outlines possible directions for future work.

2. Centered Collision Optimizer

The centered collision optimizer (CCO) is a population-based metaheuristic algorithm that draws inspiration from the physical laws governing particle collisions. In classical mechanics, the interaction between colliding bodies is governed by the conservation of momentum and the exchange of kinetic energy during impact. By interpreting candidate solutions as colliding particles within a multidimensional search domain, CCO models the optimization process through collision-driven interactions that guide the population toward regions of improved objective values [35]. Within this framework, each candidate solution is represented as a particle characterized by its position and motion within the search space. During the iterative optimization process, particles interact through collision-inspired operators that mimic different physical collision behaviors, such as elastic and partially elastic interactions. These interactions alter the movement patterns of the particles, enabling the algorithm to explore unexplored regions of the search domain while gradually refining promising candidate solutions.

A distinctive feature of CCO is the centered collision mechanism, which regulates the interaction dynamics among particles. Instead of relying solely on random pairwise interactions, the algorithm employs a centered collision mechanism to regulate the interaction dynamics among candidate solutions. By directing particle movements relative to this reference structure, the search process is guided toward more promising areas of the solution space while preserving population diversity. This mechanism contributes to maintaining an effective balance between global exploration and local exploitation throughout the optimization procedure. In addition, CCO incorporates a transformation between the original search space and a decorrelated representation of the variables. This transformation reduces the dependency among decision variables and improves the ability of the algorithm to explore complex search landscapes. Through the allocation of candidate solutions within this transformed space, the algorithm can more effectively distribute particles across the search domain and enhance the stability of the optimization process. The detailed mathematical formulation of the CCO algorithm, including the collision dynamics and transformation procedures, is provided in Appendix A for completeness.

3. Proposed Hybrid Centered Collision Optimizer with Differential Evolution

3.1. Differential Evolution Mechanism

Differential evolution (DE) is a population-based evolutionary optimization method that generates new candidate solutions through mutation, crossover, and selection operators. In the proposed hybrid framework, the classical DE operators are employed after the collision-based update stage of CCO to improve population diversity and strengthen the local refinement capability of the search process.

Mutation: Let $x_{r_1}^{(t)}$, $x_{r_2}^{(t)}$ and $x_{r_3}^{(t)}$ denote three mutually distinct individuals randomly selected from the current population at iteration t, where $r_1 \neq r_2 \neq r_3 \neq i$. In the classical DE mutation strategy, the mutant vector ($v_i^{(t)}$) is generated as:

$$v_i^{(t)} = x_{r_1}^{(t)} + F\left(x_{r_2}^{(t)} - x_{r_3}^{(t)}\right) \tag{1}$$

where $F \in (0, 2)$ is the mutation scaling factor that controls the amplification of the differential variation. Instead of assigning a fixed value to this parameter, the scaling factor is treated in this study as a stochastic quantity generated as $F = 2 \times rand(0, 1)$, where $rand(0, 1)$ denotes a uniformly distributed random number that is regenerated for every mutation operation. The amplification of the differential term therefore varies from one

individual to another and from one iteration to the next, rather than remaining fixed at a single point of the admissible interval.

Crossover: The trial vector $u_i^{(t)} = \left[u_{i,1}^{(t)}, u_{i,2}^{(t)}, \ldots, u_{i,Dim}^{(t)}\right]$ is constructed by combining the target vector $x_i^{(t)}$ and the mutant vector $v_i^{(t)}$ according to the binomial crossover rule (CR):

$$u_{i,j}^{(t)} = \begin{cases} v_{i,j}^{(t)}, & \text{if } rand_j \leq CR \text{ or } j = j_{rand} \\ x_{i,j}^{(t)}, & otherwise \end{cases} \quad (2)$$

where $CR \in [0, 1]$ is the crossover rate, $rand_j$ is a uniformly distributed random number in $[0, 1]$ and j_{rand} is a randomly selected dimension index to ensure that at least one component is inherited from the mutant vector. Throughout the present work the crossover rate is kept constant at $CR = 0.2$, so that the trial vector predominantly retains the components of the target solution obtained from the collision-based update, while a limited number of coordinates are replaced by the mutant components. The sensitivity of h-CCODE to this setting was examined on the cruise control tuning problem for $CR \in \{0.1, 0.2, 0.5\}$ (Supplementary Material). Over this range the mean tuning cost varied by less than 0.3%, the lowest mean being obtained for CR = 0.2, and it remained well below the values obtained by the other algorithms considered. A low crossover rate in the range 0.1–0.2 is therefore recommended, with CR = 0.2 as the default value.

Selection: Following the crossover stage, the trial vector is evaluated using the objective function. A greedy selection mechanism is then applied to determine the individual that survives to the next iteration:

$$x_i^{(t+1)} = \begin{cases} u_i^{(t)}, & \text{if } f\left(u_i^{(t)}\right) < f\left(x_i^{(t)}\right) \\ x_i^{(t)}, & otherwise \end{cases} \quad (3)$$

where $f(\cdot)$ denotes the objective function. This selection strategy ensures that the solution quality does not deteriorate from one iteration to the next. In the proposed h-CCODE framework, these DE operators are not used as a standalone optimizer. Instead, they are incorporated as a refinement mechanism following the collision-based update stage of CCO. In this way, the exploration capability induced by the collision dynamics is complemented by the variation and recombination ability of DE, leading to a more effective search process.

3.2. Hybridization Strategy of the Proposed H-CCODE

Although the CCO provides an effective population-based search mechanism through collision-inspired interactions among candidate solutions, its performance can still be influenced by the balance between exploration and exploitation during the optimization process. In complex search landscapes, maintaining sufficient population diversity while achieving stable convergence remains a common challenge for many metaheuristic algorithms. For this reason, hybridization strategies that incorporate complementary search operators have increasingly attracted attention in recent optimization studies [38].

Among these strategies, the integration of differential evolution (DE) operators into existing population-based algorithms has shown promising improvements in global optimization performance. The mutation and crossover operators of DE are known for their ability to generate diverse trial solutions by combining information from multiple population members. Several recent studies have demonstrated that incorporating DE mechanisms into other optimizers can significantly improve exploration capability and solution refinement. For example, the integration of DE into the gazelle optimizer has been reported to enhance convergence behavior in engineering optimization tasks [29]. Similar hybridization strategies have also been proposed for coati optimization [30] and

crayfish optimization [31], where DE mutation and crossover operators were employed to strengthen the search capability of the original algorithms. More recently, DE-assisted hybrid frameworks have been successfully applied to nonlinear process control and chaotic system identification problems, further demonstrating the effectiveness of such hybrid optimization strategies [32,33].

Motivated by these observations, this study proposes a hybrid optimization framework in which the collision-based search dynamics of CCO are combined with the evolutionary variation operators of differential evolution. The objective of this hybridization is to enhance the diversity of candidate solutions and improve the refinement capability of the search process while preserving the fundamental structure of the original CCO algorithm.

In the proposed hybrid centered collision optimizer with differential evolution (h-CCODE), the search process begins with the standard CCO update mechanism described in Section 2. Let x_i^t denote the position of the i-th candidate solution at iteration t. After the collision-based interaction stage of CCO, an intermediate candidate solution $x_i^{t,CCO}$ is obtained. To further enhance the search capability, DE-based evolutionary operators are subsequently applied to generate a trial solution $x_i^{t,DE}$. In general form, this hybrid update can be expressed as:

$$x_i^{t+1} = \begin{cases} x_i^{t,DE}, & if \ f\left(x_i^{t,DE}\right) < f\left(x_i^{t,CCO}\right) \\ x_i^{t,CCO}, & otherwise \end{cases} \tag{4}$$

where $f(\cdot)$ represents the objective function. Through this mechanism, the DE operators generate alternative candidate solutions based on the current population, and a greedy selection strategy retains the better solution for the next iteration.

It should be emphasized that the proposed hybridization does not alter the fundamental collision dynamics of the original CCO algorithm. Instead, the mutation and crossover operators of differential evolution are incorporated as an additional refinement stage following the CCO update. This design allows the algorithm to preserve the exploration capability provided by the collision-driven search mechanism while benefiting from the variation and recombination operators of DE.

The compensation expected from the DE operators is not a generic gain in diversity but follows from three properties that are complementary to the way candidate solutions are displaced in CCO. First, the collision-based update is reference-directed, in the sense that particle movements are organized relative to the centered collision structure described in Section 2, so that the resulting displacements become increasingly aligned as the population contracts around a common region. The mutation operator of Equation (1), by contrast, constructs its search direction from the difference between two randomly selected members of the population and is therefore independent of any reference point, which allows perturbations to be generated along directions that the collision mechanism can no longer produce once the population has concentrated. Second, the magnitude of this perturbation is proportional to the current dispersion of the population rather than being prescribed a priori, so the differential term is self-scaling: it remains large while the population is spread and contracts automatically as the search settles, and the stochastic factor $F = 2 \times rand(0,1)$ prevents this contraction from becoming absolute by admitting occasional steps of up to twice the sampled difference. This property acts directly on the stagnation mode identified above, in which the baseline algorithm loses the ability to escape a basin once its population has collapsed into it. Third, the binomial crossover of Equation (2) recombines the target and mutant vectors coordinate by coordinate, whereas the collision update displaces each candidate as a whole; partial recombination therefore produces solutions that are not reachable by the collision dynamics alone and complements, at the level of individual variables,

the decorrelation transformation that CCO applies at the level of the representation. The greedy selection of Equation (3) completes this arrangement by ensuring that none of the additional variation can degrade the population, since a trial solution replaces its parent only upon improvement, which is the mechanism that allows the baseline behavior to be preserved on the functions where CCO already attains the optimum.

The distinction between the present framework and existing DE-assisted hybrids, such as those built on the gazelle [29], coati [30], and crayfish [31] optimizers, should also be made explicit, since the combination of DE with a population-based optimizer is by itself an established practice. Three aspects differentiate the scheme proposed here. The first concerns the structure of the base algorithm: CCO maintains two representations of the population simultaneously, an original and a decorrelated one, with an allocation ratio that evolves during the run, and the DE stage is deliberately positioned after the decorrelated-space individuals have been mapped back, so that the differential operators act on a population carrying information from both representations. Such a configuration does not arise in the single-representation frameworks of [29–31], where the DE stage necessarily operates on the same representation that produced the candidates. The second concerns the strength assigned to the DE component. Rather than introducing a full differential evolution search alongside the base optimizer, the operators are configured as a weak refinement through the low crossover rate discussed in Section 3.1, so that the collision dynamics remain the primary search driver and the hybrid does not degenerate into a DE variant in which the base algorithm plays a secondary role. The benchmark results support this design, as the baseline performance is reproduced exactly on the functions where CCO already converges to the optimum. The third concerns the motivation itself: the modification reported here was introduced in response to deficiencies that are identified and quantified for this particular base algorithm in Section 4, and is accompanied by an application to controller tuning, a problem class to which neither CCO nor any DE-enhanced form of it had previously been applied. The novelty claimed in this study is accordingly that of a targeted and empirically justified structural enhancement together with its control-oriented validation, and not that of a new search paradigm.

The classical mutation and crossover operators of differential evolution employed in this work follow the standard DE formulation introduced in Section 3.1, namely Equations (1)–(3), and no additional operator is introduced within the hybrid scheme. For clarity and reproducibility of the proposed method, the complete algorithmic procedure of the hybrid optimizer is summarized in Algorithm 1.

Algorithm 1. Pseudo-code of h-CCODE

Require: Population size P, lower and upper bounds x_l and x_u, problem dimension Dim,
maximum iteration number T_{max}, mutation factor F, crossover rate CR,
initial values of kp_{OS} and kp_{DS} set to 0.5.
Ensure: Best solution x_{best} and its fitness F_{best}.
Initialize the population x within the search bounds.
Evaluate the fitness of all individuals.
Determine the best solution x_{best} and its fitness F_{best}.
Set $t = 1$.
while $t < T_{max}$ **do**
    Sort the population according to fitness values.
    Update x_{best} and F_{best}.

Algorithm 1. *Cont.*

Evolve a subset of individuals in the original space using the centered collision-based update mechanism strategy.

Transform another subset into the decorrelated space and update them based on the decorrelation transformation.

Map the updated decorrelated-space individuals back to the original space.

Apply DE mutation to generate mutant vectors for each individual.

Apply DE crossover to generate trial vectors.

Evaluate the trial vectors.

Perform greedy selection for each individual between the current solution and the corresponding trial solution.

Update the allocation ratios of the original and decorrelated spaces.

Check and correct boundary violations.

Re-evaluate the population if required.

Update x_{best} and F_{best}.

Set $t = t + 1$.

end while

Return x_{best} and F_{best}.

4. Benchmark Evaluation on the CEC2022 Test Suite

To further evaluate the performance of the proposed h-CCODE, additional experiments were carried out using the CEC2022 single-objective bound-constrained benchmark functions [39]. This benchmark suite is widely used for assessing metaheuristic optimization algorithms due to its diverse structure and varying levels of difficulty. It consists of 12 minimization problems, including unimodal, multimodal, hybrid, and composition functions, which enable a comprehensive evaluation of both exploration and exploitation capabilities under different landscape characteristics. In all experiments, the search domain was defined as $[-100, 100]^D$, and the benchmark functions were handled in accordance with standard evaluation procedures. A summary of the adopted test functions is presented in Table 1.

Table 1. Summary of IEEE CEC2022 benchmarks functions.

ID	Function Type	D	f_{min}
F1	Unimodal	10	300
F2	Multimodal	10	400
F3		10	600
F4		10	800
F5		10	900
F6	Hybrid	10	1800
F7		10	2000
F8		10	2200
F9	Composition	10	2300
F10		10	2400
F11		10	2600
F12		10	2700
Search range: $[-100,\ 100]^D$			

In order to ensure a fair comparison, the original CCO and the proposed h-CCODE were tested under identical experimental conditions. For both algorithms the population size was set to 50 and the search was terminated after a maximum of 1000 iterations, which corresponds to a budget of 50,000 function evaluations per run and is consistent with the stopping criteria commonly adopted for this benchmark suite. The DE stage of h-CCODE was operated with the parameter setting introduced in Section 3.1, that is, $CR = 0.2$ and $F = 2 \times rand(0, 1)$. Each algorithm was executed over 30 independent runs on every test function, and all reported statistical indicators, namely the mean, standard deviation, best, and worst objective values, were computed from these 30 runs. In addition to these descriptive measures, the Wilcoxon signed-rank test was applied at a significance level of 0.05 in order to establish whether the observed differences between the two algorithms are statistically meaningful rather than a consequence of the stochastic nature of the search.

It should be noted that the comparison presented in this section is focused on the original CCO, since the question addressed here is the extent to which the DE-based refinement stage modifies the behavior of the baseline optimizer, and this effect can only be isolated through a direct comparison under identical conditions. The baseline algorithm itself has been benchmarked extensively in its foundational study [34], where its superiority over a broad set of classical and parameter-free metaheuristics, including particle swarm optimization (PSO) [40] and the grey wolf optimizer (GWO) [41], was established. In order to position h-CCODE also with respect to other optimizers, the CEC2022 experiment was repeated under the settings described above with PSO and GWO as well-established metaheuristics and with the mountain gazelle optimizer (MGO) [42] and quadratic interpolation optimization (QIO) [43] as recent algorithms that require no algorithm-specific parameters. The results are reported in Section S1.2 and Table S2 of the Supplementary Material. h-CCODE attains the lowest mean value on all twelve functions, the result being shared with CCO, and in two cases also with MGO, on the functions where several algorithms reach the optimum, and it obtains the best average rank (1.25), followed by CCO (2.08), QIO (3.54), MGO (3.58), PSO (4.92), and GWO (5.62). The second position of CCO in this ranking is consistent with the comparisons reported in [34].

The numerical results obtained from the benchmark experiments are reported in Table 2. As can be seen from the table, both algorithms yield identical or nearly identical results for relatively simple functions such as F1, F3, and F5, where the obtained objective values coincide with the known optimum values. This observation suggests that the incorporation of DE operators does not degrade the baseline performance of the original CCO in simpler search landscapes.

More noticeable differences can be observed in functions with higher complexity, particularly those involving multimodal, hybrid, and composition structures. In these cases, h-CCODE generally achieves lower mean objective values compared to CCO, indicating an improved capability to avoid local optima. In addition, the standard deviation values are often smaller, suggesting a more stable search behavior across independent runs. The improvement is especially evident in functions such as F2, F4, F6, F7, F10, and F11, where h-CCODE provides better average performance than the original algorithm. For example, in F10 and F11, the difference in mean values is more pronounced, reflecting the contribution of the DE-based mutation and crossover operators to the global search process. Although the improvements obtained for some functions, such as F8 and F12, are smaller in magnitude, the overall tendency still favors the proposed method.

Furthermore, for functions such as F9, both algorithms yield nearly identical results, indicating that the hybrid structure preserves the original behavior of CCO when the problem does not strongly challenge the search process. This balance between maintaining

baseline performance and improving results on more complex functions demonstrates the effectiveness of the proposed hybrid approach.

Table 2. Comparison results of h-CCODE and CCO.

ID	Algorithm	Mean	SD	Best	Worst
F1	h-CCODE	3.0000×10^2	0	3.0000×10^2	3.0000×10^2
	CCO	3.0000×10^2	0	3.0000×10^2	3.0000×10^2
F2	h-CCODE	4.0000×10^2	0	4.0000×10^2	4.0000×10^2
	CCO	4.0254×10^2	3.3391	4.0000×10^2	4.0892×10^2
F3	h-CCODE	6.0000×10^2	0	6.0000×10^2	6.0000×10^2
	CCO	6.0000×10^2	9.7469×10^{-6}	6.0000×10^2	6.0000×10^2
F4	h-CCODE	8.0418×10^2	1.2231	8.0199×10^2	8.0497×10^2
	CCO	8.0665×10^2	2.6901	8.0298×10^2	8.1293×10^2
F5	h-CCODE	9.0000×10^2	0	9.0000×10^2	9.0000×10^2
	CCO	9.0000×10^2	6.1398×10^{-14}	9.0000×10^2	9.0000×10^2
F6	h-CCODE	1.8001×10^3	4.4458×10^{-2}	1.8000×10^3	1.8001×10^3
	CCO	1.8005×10^3	6.8406×10^{-1}	1.8000×10^3	1.8021×10^3
F7	h-CCODE	2.0001×10^3	2.6314×10^{-1}	2.0000×10^3	2.0006×10^3
	CCO	2.0034×10^3	6.5760	2.0000×10^3	2.0220×10^3
F8	h-CCODE	2.2046×10^3	8.0585	2.2000×10^3	2.2208×10^3
	CCO	2.2106×10^3	9.7237	2.2005×10^3	2.2213×10^3
F9	h-CCODE	2.5293×10^3	9.2825×10^{-14}	2.5293×10^3	2.5293×10^3
	CCO	2.5293×10^3	8.4057×10^{-13}	2.5293×10^3	2.5293×10^3
F10	h-CCODE	2.4044×10^3	2.0020×10	2.4000×10^3	2.5004×10^3
	CCO	2.5045×10^3	2.1328×10	2.5002×10^3	2.6069×10^3
F11	h-CCODE	2.6000×10^3	1.3127×10^{-13}	2.6000×10^3	2.6000×10^3
	CCO	2.6060×10^3	3.0085×10	2.6000×10^3	2.7504×10^3
F12	h-CCODE	2.8630×10^3	1.5878	2.8588×10^3	2.8640×10^3
	CCO	2.8648×10^3	5.1810×10^{-1}	2.8627×10^3	2.8650×10^3

Read together with the best and worst values of Table 2, these results also identify the conditions under which the baseline algorithm is deficient, and the two forms this takes are distinct. On F2, F7, and F11 the original CCO reaches the known optimum in its best run yet fails to do so consistently, the worst results over the 30 runs deteriorating to 408.92, 2022.0, and 2750.4 against optimum values of 400, 2000, and 2600, respectively; the global basin is therefore located but not retained, which is the signature of a population that collapses prematurely in a subset of the runs. The corresponding standard deviations of 3.34, 6.58, and 30.09 quantify the severity of this inconsistency, and all three are reduced to zero or to the order of numerical precision by the proposed hybrid. On F10 the difficulty is of a different nature, since the best value attained by CCO in any of the 30 runs is 2500.24 while the optimum lies at 2400; the population settles in an inferior basin in every run, so that the failure is one of exploration rather than of convergence, and h-CCODE reaches 2400.00 in its best run and reduces the mean error above the optimum from 104.50 to 4.40. Expressed in the same terms, the mean error is reduced by 97.1% on F7, 95.8% on F10, 80.0% on F6, and 56.6% on F8, and is eliminated entirely on F2 and F11. Equally relevant is where no deficiency exists: on the unimodal function F1 and on F3, F5, and F9 both algorithms converge to the same values, so the added operators neither improve nor disturb a search that is already adequate. The advantage obtained by hybridization is thus confined to rugged and multi-basin landscapes, which is consistent with the mechanisms set out in Section 3.2, and it is obtained predominantly through improved consistency across runs rather than through a higher peak solution quality. For completeness it should be noted that F12 constitutes the single exception to this pattern, the mean being improved while the standard deviation increases from 0.52 to 1.59, which indicates that the additional variation is not uniformly beneficial on every composition landscape.

These observations are supported by the outcome of the Wilcoxon signed-rank test carried out over the 30 independent runs. For F1, F3, F5, and F9 the two algorithms converged to identical or practically identical objective values, so that the test returns no significant difference and the corresponding performances must be regarded as statistically equivalent. For the remaining eight functions, that is F2, F4, F6, F7, F8, F10, F11, and F12, the computed p-values fall below the 0.05 significance level, with most of them below 0.01, which confirms that the improvement attained by h-CCODE on the more demanding landscapes is statistically significant and not attributable to random variation between runs.

5. Real-World Engineering Application: Vehicle Cruise Control

5.1. Mathematical Modeling of Test System

The proposed control approach is assessed on a vehicle cruise control system, which is frequently used as a benchmark example in control studies due to its nonlinear longitudinal dynamics and practical relevance. The objective of the cruise control system is to regulate the vehicle speed by adjusting the throttle input in response to a reference velocity. The longitudinal dynamics of the vehicle are affected by aerodynamic resistance, road slope, actuator dynamics, and external disturbances. A schematic representation of the nonlinear cruise control system is shown in Figure 1.

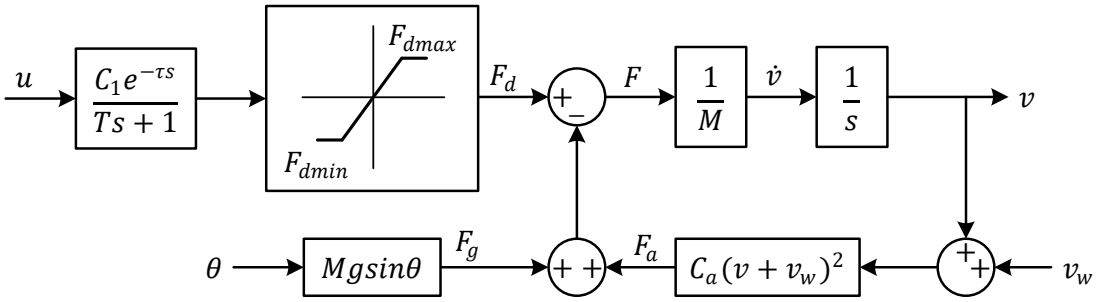


Figure 1. Nonlinear vehicle cruise control model with actuator, aerodynamic drag, and road-slope effects.

The longitudinal motion of the vehicle can be described based on Newton's second law as follows:

$$F_d = M\frac{dv}{dt} + F_a + F_g \tag{5}$$

where F_d denotes the driving force generated by the actuator, F_a represents the aerodynamic drag force, and F_g is the gravitational force due to road inclination. The aerodynamic drag is modeled as a quadratic function of the relative velocity, and the climbing resistance is expressed as:

$$F_a = C_a(v + v_w)^2, F_g = Mg\sin\theta \tag{6}$$

where C_a is the aerodynamic drag coefficient, v is the vehicle velocity, v_w denotes wind disturbance, M is the vehicle mass, and θ is the road slope. The actuator dynamics of the cruise control system are typically modeled as a first-order system with a time delay, given by:

$$F_d(s) = \frac{C_1 e^{-\tau s}}{Ts+1}u(s), \quad F_{d\,min} \le F_d \le F_{d\,max} \tag{7}$$

where u is the control input, C_1 is the actuator gain, T is the time constant, and τ represents the time delay. By combining the above relationships, the nonlinear dynamic model of the system can be expressed in state form as:

$$\dot{v} = \frac{1}{M}\left(F_d - C_a v^2\right), \quad \dot{F}_d = \frac{1}{T}(C_1 u(t-\tau) - F_d) \tag{8}$$

It should be noted that the road-slope and wind terms of Equation (6) do not appear in Equation (8), which describes the vehicle travelling on a level road in still air. These two quantities are not part of the model used for controller synthesis; they are treated as external disturbances acting on the closed loop. For controller design, the nonlinear system is linearized around a nominal operating point v_0. Considering small perturbations around this equilibrium, the linearized model can be written as:

$$\delta\dot{v} = -\frac{2C_a v_0}{M}\delta v + \frac{1}{M}\delta F_d, \quad \delta\dot{F}_d = -\frac{1}{T}\delta F_d + \frac{C_1}{T}\delta u(t-\tau) \tag{9}$$

The linearization in Equation (9) is performed about the steady state at which the driving force balances the aerodynamic resistance, the quadratic drag term being replaced by its first-order expansion about v_0, which gives rise to the coefficient $2C_a v_0/M$. Applying the Laplace transform to Equation (9) with zero initial conditions yields two first-order transfer functions connected in series. Their product gives the plant from throttle input to speed deviation. To obtain a rational model suitable for simulation-based tuning, the delay is replaced by the first-order lag which is adequate over the frequency range of interest because the closed-loop bandwidth reported in Section 6.5 lies below the corner frequency $1/\tau = 5\ rad/s$. The plant then assumes the third-order form of Equation (10). For the benchmark model adopted in previous studies [1], these expressions give the open-loop transfer function:

$$G_p(s) = \frac{a_0}{(s-s_1)(s-s_2)(s-s_3)} = \frac{2.4767}{(s+0.0476)(s+1)(s+5)} \tag{10}$$

This transfer function captures the main dynamic characteristics of the cruise control system and has been used as a benchmark plant model in a number of cruise control studies.

5.2. Closed-Loop Structure and PID Controller Design

In order to regulate the vehicle speed and track the desired reference signal, a closed-loop control structure is adopted. Based on the linearized plant model derived in Section 5.1, a feedback control framework is constructed. The overall closed-loop configuration of the cruise control system, including external disturbances and measurement noise, is illustrated in Figure 2.

In this structure, the reference velocity signal is compared with the measured vehicle speed to generate the tracking error. This error is then processed by the controller to produce the control input $u(t)$, which drives the actuator. The plant dynamics are influenced not only by the control input but also by external disturbances, such as wind effects and road conditions. In addition, measurement noise is considered in the feedback path to provide a more realistic representation of practical operating conditions. In this study, PID controller is employed due to its simplicity, robustness, and widespread use in industrial control applications. The controller is defined in the Laplace domain as:

$$C(s) = K_P + \frac{K_I}{s} + K_D s \tag{11}$$

where K_P, K_I and K_D denote the proportional, integral, and derivative gains, respectively. The proportional term provides an immediate response to the tracking error, the integral term eliminates steady-state error, and the derivative term improves the transient response by anticipating system behavior. Although PI-based structures have been considered in several cruise control studies, a PID controller is adopted here to provide additional flexibility in shaping the transient response. The closed-loop system shown in Figure 2 allows the evaluation of the controller performance under realistic conditions, where both disturbance inputs and measurement noise are present. This framework forms the basis for the optimization process described in the following section, in which the controller parameters are tuned using the proposed h-CCODE algorithm.

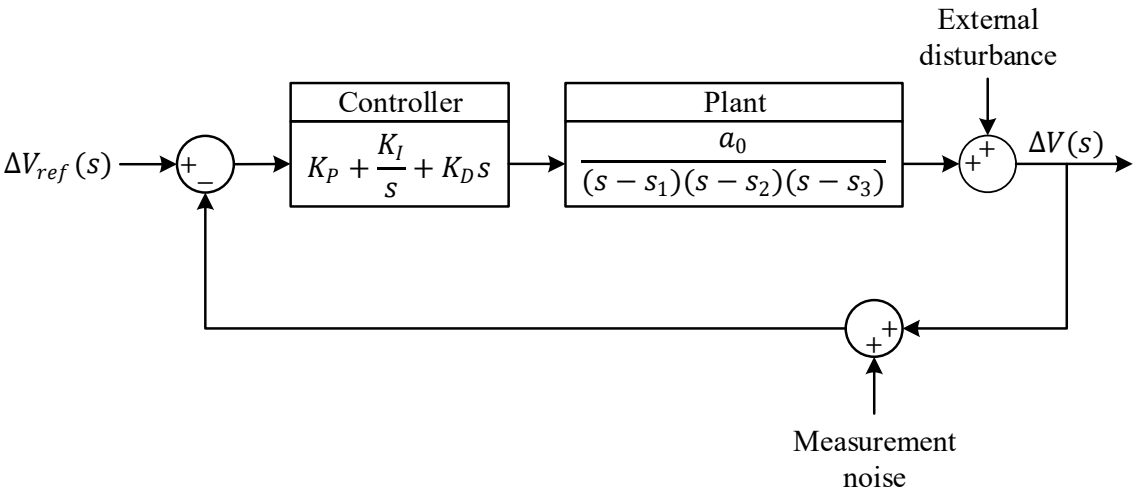


Figure 2. Closed-loop PID-based vehicle cruise control system with external disturbance and measurement noise.

5.3. Optimization Problem Formulation

In order to achieve an effective speed regulation performance, the tuning of the PID controller parameters is formulated as an optimization problem. The objective is to determine the optimal values of K_P, K_I and K_D such that the closed-loop system exhibits a desirable transient and steady-state response. To this end, a composite performance index is defined by combining several time-domain specifications. The proposed cost function considers the percent overshoot, settling time, and the integral of absolute error, and is expressed as:

$$Cost = \theta_1 \times p_{os} + \theta_2 \times t_{set} + (1 - \theta_1 - \theta_2) \times \int_0^{t_f} |e(t)| dt \tag{12}$$

where p_{os} represents the percent overshoot, t_{set} denotes the settling time defined within a $\pm 2\%$ tolerance band, and $e(t)$ is the tracking error between the reference and actual vehicle speed. The final time of simulation is selected as $t_f = 5$ s. The weighting factors are chosen as $\theta_1 = 0.1$ and $\theta_2 = 0.4$, which provide a balanced trade-off between transient response and steady-state performance. In this formulation, the integral of absolute error contributes significantly to the cost function, ensuring accurate tracking over the entire simulation horizon, while the inclusion of overshoot and settling time helps to control undesirable transient characteristics.

It should be noted that the quantities combined in Equation (12) are evaluated within a normalized framework, which removes the scaling conflict that would otherwise arise between terms expressed in different physical units. The simulated responses reflect these normalized dynamics. Absolute physical values are recovered by rescaling the normalized

outputs with the corresponding nominal operating values, for instance, a reference speed of 30 km/h. The weighting factors were likewise not selected arbitrarily; they were established through preliminary closed-loop response analyses in which the relative contribution of each term was varied until an acceptable compromise between transient behavior and steady-state accuracy was obtained, the retained values being those that produced the most balanced response in terms of overshoot, settling time, and tracking error simultaneously.

The optimization problem is subject to practical bounds on the controller parameters, defined as: $1 \leq K_P \leq 5$, $0.1 \leq K_I \leq 0.5$ and $1 \leq K_D \leq 8$. These bounds are selected to restrict the search space to practically reasonable controller values. In this study, the optimal PID parameters are obtained by minimizing the defined cost function using the proposed h-CCODE algorithm. The optimization process is carried out on the closed-loop system described in Section 5.2, where the influence of disturbances and measurement noise is also taken into account.

6. Simulation Results and Performance Evaluation

The performance of the proposed h-CCODE-based PID controller is evaluated through a series of simulation studies carried out in MATLAB/Simulink R2023b under identical conditions. The assessment is conducted from multiple perspectives, including optimization performance, closed-loop time-domain behavior, disturbance rejection capability, and robustness analysis. Unless otherwise stated, all closed-loop simulations were performed with the fixed-step ode4 (Runge–Kutta) solver using a fundamental step size of 0.001 s and a total simulation duration of 15 s, whereas the cost function in Equation (12) is accumulated over the horizon t_f specified in Section 5.3. In addition to the comparison with the original CCO algorithm, the proposed approach is also evaluated against several methods reported in the literature, including classical, popular, and recently developed optimization techniques such as GA [18], ALO [19], ASO [20], RSALF [21], AOA-NM [23], and SMA [24]. These comparisons provide a broader perspective on the effectiveness of the proposed method. In addition, a controlled comparison in which PSO, GWO, MGO, and QIO are applied to the same tuning problem under conditions identical to those of h-CCODE and CCO is reported in Section S1.3 of the Supplementary Material and summarized in Section 6.1.

6.1. Optimization Performance Analysis

To evaluate the effectiveness of the proposed optimization strategy, both h-CCODE and the original CCO algorithm were executed over 30 independent runs with a population size of 20 and a stopping criterion defined by a maximum of 50 iterations, corresponding to 1000 function evaluations per run. The DE stage was operated with the same parameter setting used in the benchmark experiments. All statistical indicators reported below were obtained from these 30 independent runs. The convergence behavior of the algorithms is illustrated in Figure 3. As can be observed, both algorithms show a rapid decrease in the cost function during the initial iterations. However, h-CCODE reaches lower cost values more quickly and maintains a more stable convergence trend compared to CCO. This behavior suggests that the integration of DE operators improves both the exploration capability and convergence stability of the original algorithm.

The statistical results of the cost function are presented in Table 3. It is evident that h-CCODE achieves a lower mean cost value than CCO, indicating improved overall solution quality. In addition, the standard deviation obtained by h-CCODE is smaller, which reflects a more consistent performance across independent runs. The best and worst values further support this observation, showing that the proposed method produces more reliable optimization outcomes. In order to verify that the difference observed between the two algorithms is statistically meaningful, the Wilcoxon signed-rank test was applied

to the cost values obtained from the 30 independent runs. The test yielded a p-value of 1.7344×10^{-6}, which is well below the significance level of 0.05 and therefore rejects the null hypothesis of equal performance. The reduction in the cost function achieved by h-CCODE can thus be attributed to the DE-based refinement stage rather than to the stochastic dispersion of the individual runs. The same tuning problem was also solved with PSO, GWO, MGO, and QIO, using the population size, iteration budget, cost function, parameter bounds, and number of independent runs given above (Supplementary Material, Section S1.3 and Table S3). The mean cost of h-CCODE is lower by 5.5–7.0% than those of these four algorithms, its standard deviation is the smallest among the six algorithms considered, and its worst result over the 30 runs (0.6703) is lower than the best results obtained by PSO, GWO, and MGO. In terms of convergence rate (Supplementary Material, Figure S1 and Table S4), h-CCODE reaches a cost of 0.70 or lower at the 8th iteration, whereas the other five algorithms require between 13 (CCO) and 34 (GWO) iterations, and by the 11th iteration it attains a cost lower than the final values reached by PSO, GWO, MGO, and QIO after 50 iterations.

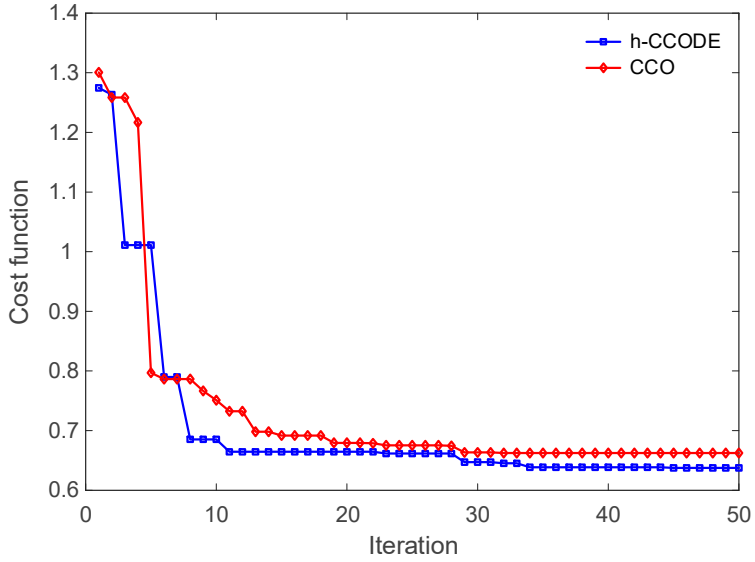


Figure 3. Convergence curves of the cost function for h-CCODE and CCO.

Table 3. Statistical results of the cost function.

Algorithm	Mean	SD	Best	Worst
h-CCODE	0.6531	0.0095	0.6375	0.6703
CCO	0.6809	0.0133	0.6625	0.7101

6.2. Closed-Loop Time-Domain Performance

The closed-loop performance of the cruise control system is evaluated in terms of time-domain response characteristics. The step response of the system for both h-CCODE and CCO-based PID controllers is shown in Figure 4. As observed from the figure, both controllers are capable of tracking the reference signal. However, the response obtained using h-CCODE exhibits a slightly faster rise and a reduced overshoot compared to the CCO-based controller. In addition, the settling behavior is smoother, indicating an improved transient response.

The quantitative performance metrics are summarized in Table 4, including rise time, settling time, and percent overshoot. The results show that the h-CCODE-based controller achieves lower settling time and significantly reduced overshoot compared to CCO, while maintaining comparable rise time.

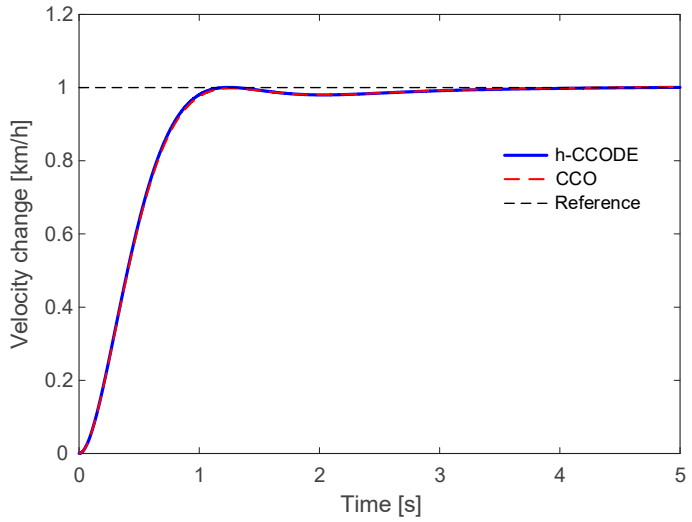


Figure 4. Closed-loop step responses of the cruise control system using h-CCODE and CCO.

Table 4. Time-domain performance metrics (rise time, settling time, and overshoot).

Algorithm	t_{rise} [s]	t_{set} [s]	p_{os} [%]
h-CCODE	0.6419	0.9968	0.0629
CCO	0.6532	1.0232	0.1810

The corresponding optimal PID parameters obtained by both algorithms are listed in Table 5. It can be observed that the parameter set generated by h-CCODE leads to improved time-domain performance, which is consistent with the observed response characteristics.

Table 5. Optimal PID parameters obtained by h-CCODE and CCO.

Algorithm	K_P	K_I	K_D
h-CCODE	4.4707	0.2355	5.2404
CCO	4.3896	0.2453	5.1693

6.3. Comparative Evaluation with Existing Methods

To further assess the effectiveness of the proposed h-CCODE-based PID controller, a comparative analysis is conducted against several optimization-based control methods reported in the literature. These include both classical and recently developed approaches such as GA, ALO, ASO, RSALF, AOA-NM, and SMA. The closed-loop responses obtained using different optimization methods are illustrated in Figure 5. It should be clarified that the controller gains of the compared methods, listed in Table 6, are those reported in the corresponding studies, whereas all closed-loop evaluations presented in this section were carried out in the present work under identical conditions. Each gain set was implemented on the same plant model of Equation (10), for which these designs were originally developed, and was simulated with the same reference input, solver settings, and simulation horizon described at the beginning of Section 6; the disturbance and noise signals of Section 6.4 were likewise applied unchanged to every controller, and all performance measures were computed through a common procedure. The comparison is therefore one between final controller designs assessed under a single set of conditions, rather than between optimization runs. This arrangement was preferred to re-implementing the competing algorithms for two reasons. First, the published gain sets represent the designs regarded as best by the respective authors under their own algorithm settings, so that each method is represented by its reported outcome rather than by an implementation and

parameterization chosen in this study, which would introduce a source of bias outside the control of the original authors.

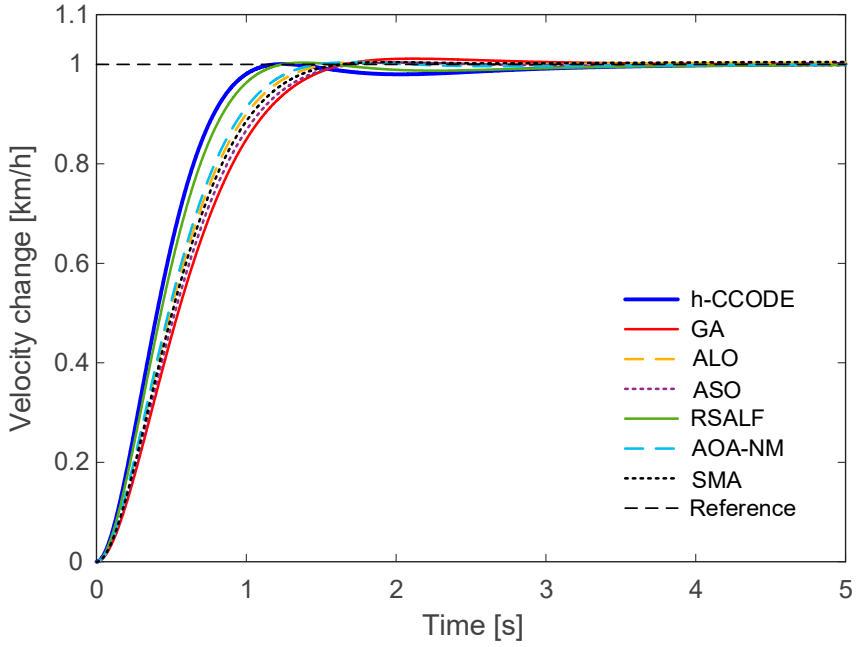


Figure 5. Closed-loop responses of the proposed method and existing optimization-based controllers.

Table 6. PID parameters obtained by different optimization methods.

Algorithm	K_P	K_I	K_D
GA	3.5907	0.1630	3.3021
ALO	3.8580	0.2089	3.8581
ASO	3.6629	0.1520	3.5233
RSALF	4.3714	0.1972	4.7421
AOA-NM	3.9840	0.1798	3.9976
SMA	3.7448	0.2098	3.7227

Second, a controlled comparison between optimizers under identical population size, iteration budget, cost function, and number of runs is provided separately, both for the pair in which the effect of the proposed modification can be isolated and tested statistically, namely h-CCODE and the original CCO (Sections 4 and 6.1), and for PSO, GWO, MGO, and QIO, which were applied to the present tuning problem under the same conditions (Supplementary Material, Section S1.3). This controlled comparison is needed because several of the earlier methods in Table 6, GA in particular, are known to return suboptimal solutions on problems of this kind, which is consistent with the GA-based design exhibiting the slowest response and the largest overshoot in Table 7. The literature designs are therefore retained as reference points for the controller quality reported for this benchmark plant, rather than as the sole basis for assessing the optimizer. It may further be noted that all gains in Table 6 lie within the parameter bounds defined in Section 5.3, so the advantage of the h-CCODE design does not stem from access to a region of the parameter space unavailable to the other designs, and that the assessment is not confined to the criteria entering Equation (12): the ISE, ITAE, and ITSE indices of Table 8 play no part in the cost function, and on ITAE the RSALF-based design slightly outperforms the proposed one. In terms of the individual experimental conditions, the plant, reference input, disturbance and noise signals, simulation time and solver, and performance measures are therefore

common to all methods, whereas the parameter bounds and objective function belong to the design stage of each method and are those adopted in the original studies.

Table 7. Time-domain performance comparison of different optimization methods.

Algorithm	t_{rise} [s]	t_{set} [s]	p_{os} [%]
h-CCODE	0.6419	0.9968	0.0629
GA	0.9392	1.4558	1.1456
ALO	0.8334	1.3092	0.4794
ASO	0.9040	1.4306	0.3677
RSALF	0.6922	1.0721	0.3553
AOA-NM	0.8036	1.2572	0.4147
SMA	0.8648	1.3664	0.4735

Table 8. Performance indices under disturbance conditions (IAE, ISE, ITAE, ITSE).

Algorithm	$IAE=\int_0^{t_f} \lvert e(t)\rvert dt$	$ISE=\int_0^{t_f} e^2(t)dt$	$ITAE=\int_0^{t_f} t\lvert e(t)\rvert dt$	$ITSE=\int_0^{t_f} te^2(t)dt$
h-CCODE	0.6672	0.3182	1.7293	0.1897
GA [18]	0.8784	0.4291	2.2627	0.2974
ALO [19]	0.7979	0.3883	2.1269	0.2562
ASO [20]	0.8394	0.4124	2.1623	0.2788
RSALF [21]	0.6909	0.3383	1.7223	0.2072
AOA-NM [23]	0.7635	0.3784	1.9286	0.2453
SMA [24]	0.8288	0.3983	2.2543	0.2665

As can be observed from the figure, all methods track the reference signal without sustained oscillation, which is consistent with the well-damped dominant closed-loop poles of all designs, whose damping ratios lie between 0.70 and 0.88. The differences between the responses therefore lie mainly in their speed and in the size of the overshoot rather than in their oscillatory character. The h-CCODE-based controller produces the fastest transition, with a closed-loop bandwidth of about 3.48 rad/s against 2.27–3.20 rad/s for the designs taken from the literature, and it combines this speed with the smallest overshoot of all methods (Table 7). Among the literature designs, the RSALF-based controller comes closest in bandwidth (about 3.20 rad/s), yet its overshoot of 0.36% is more than five times that of the proposed design, while the GA-based response is the slowest and at the same time exhibits the largest overshoot, 1.15%, before settling.

The corresponding PID parameters obtained by each optimization method are listed in Table 6. It can be seen that different algorithms produce varying parameter combinations, which directly influence the dynamic response of the system.

To provide a quantitative comparison, the time-domain performance metrics are summarized in Table 7, including rise time, settling time, and percent overshoot. From Table 7, it is evident that the proposed h-CCODE-based controller achieves the lowest rise time, settling time, and overshoot among the compared methods. These results indicate that the proposed approach provides the most favorable overall transient response. Although several methods exhibit acceptable tracking behavior, the h-CCODE-based design offers a more balanced and improved closed-loop performance.

6.4. Disturbance Rejection Performance

To further evaluate the practical applicability of the proposed controller, disturbance rejection performance is investigated by introducing external disturbances and measurement noise into the system. The disturbance and noise signals applied to the system are illustrated in Figure 6. The amplitudes of both signals are expressed in km/h. The external

disturbance is introduced as a step-type signal producing sudden speed variations of ±0.1 km/h at predefined instants, while the measurement noise is generated by a band-limited white noise source with a noise power of 0.000005 and a sample time of 0.05 s. Although the disturbance of Figure 6 is specified as a sequence of steps, it is intended to represent the effect on vehicle speed of the physical loads included in the model rather than an arbitrary test input. The two external loads in Equation (6) are the climbing resistance associated with the road slope θ and the aerodynamic drag associated with the wind speed, and both change abruptly in common driving situations: the gravitational component changes in a step-like manner when the vehicle moves onto a gradient, and the relative air speed changes almost instantaneously when a headwind gust is encountered or when the vehicle leaves the shelter of a tunnel, a cutting, or a larger vehicle. The step is also the most demanding deterministic form that such a load can take. For a linear closed loop, a disturbance that reaches the same level through a gradual, monotonic onset, as in a progressive change of grade, produces a response that is a weighted average of the step response and therefore cannot exhibit a larger peak deviation. Reversing the sign of the disturbance, as in Figure 6, subjects the loop to both uphill-type and downhill-type loads within a single test, while the measurement noise accounts for the uncertainty introduced by the speed sensor. Standard driving cycles were not adopted for this purpose, since they prescribe a target speed profile rather than a disturbance profile and are designed for the assessment of energy consumption and emissions over urban and extra-urban operation, including stop-and-go segments that lie outside the envelope in which cruise control is normally engaged. Their role in the present context is closer to that of the reference tracking test in Section 6.5, in which the controller is required to follow successive positive and negative speed changes.

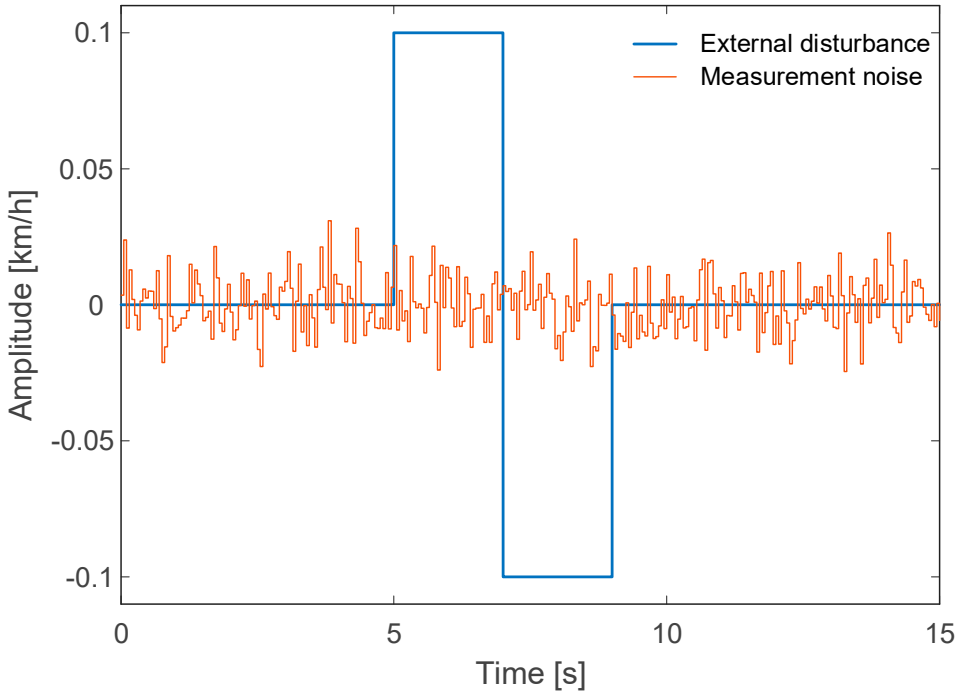


Figure 6. External disturbance and measurement noise signals applied to the system.

The corresponding system responses under these conditions are presented in Figure 7. As can be observed from Figure 7, the proposed h-CCODE-based controller is capable of maintaining stable operation in the presence of disturbances. The system response remains well-behaved, and deviations caused by disturbances are attenuated within a short

time interval. Compared to the reference signal, the output exhibits limited oscillations, indicating effective disturbance suppression.

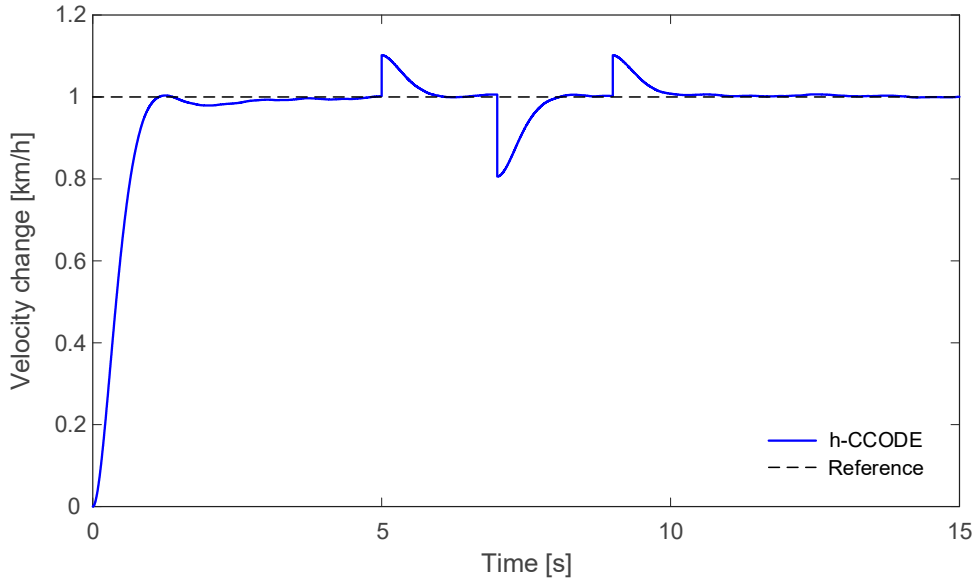


Figure 7. Disturbance rejection performance of the proposed h-CCODE-based PID controller.

To quantify the disturbance rejection capability, several performance indices are calculated, including the integral of absolute error (IAE), integral of squared error (ISE), integral of time-weighted absolute error (ITAE), and integral of time-weighted squared error (ITSE). The results are summarized in Table 8. From Table 8, it can be seen that the proposed controller performs favorably across all considered disturbance-related performance indices. In particular, h-CCODE provides the lowest values for IAE, ISE, and ITSE, while its ITAE result remains very close to the best-performing method. These findings suggest that the proposed control strategy is effective in limiting error growth and preserving satisfactory disturbance rejection behavior under non-ideal operating conditions.

6.5. Robustness and Stability Analysis

In addition to time-domain performance evaluation, the stability and robustness of the proposed control system are further investigated under different operating conditions. The frequency-domain characteristics of the system are first analyzed using the open-loop Bode diagram shown in Figure 8. The obtained frequency response yields a phase margin of 69.5786°, an infinite gain margin, and a bandwidth of 3.4791 rad/s. These results indicate that the system possesses satisfactory stability margins and exhibits a sufficiently responsive behavior in the frequency domain. To further evaluate the tracking capability of the controller, a reference signal with successive positive and negative variations is applied. The corresponding system response over a 25 s interval is presented in Figure 9. For the CCO-based controller, the same analysis gives a phase margin of approximately 69.98°, an infinite gain margin, and a bandwidth of about 3.43 rad/s. The stability margins of the two designs are therefore practically identical, and the differences between them reside in the time-domain characteristics reported in Tables 4 and 7.

As observed from Figure 9, the proposed controller is able to follow the varying reference signal with limited tracking error and without sustained oscillations. The response remains stable throughout the entire simulation period, indicating that the controller can handle changing operating conditions effectively.

The robustness of the control system is further examined by introducing parameter uncertainties into the model. The closed-loop responses obtained under nominal conditions

and four different uncertainty scenarios are illustrated in Figure 10. The two perturbed parameters and their ranges were selected on physical grounds. The vehicle mass was varied between 1350 and 1600 kg, that is from −10% to +6.7% of its nominal value, which corresponds to the change in loading that a passenger car undergoes in ordinary service as the number of occupants, the luggage carried, and the fuel level vary. The actuator delay was varied over a wider interval, between 0.18 and 0.25 s, or from −10% to +25% of its nominal value, for two reasons. The delay lumps together the throttle response and the build-up of engine torque, both of which depend on engine speed and gear selection and are therefore known less precisely than the mass; in addition, an increase in delay is the direction in which the phase margin of the loop is eroded, so the range was deliberately extended further on that side. The four cases were arranged so that every combination of sign is represented, namely a lighter vehicle with a slower actuator (Case I), a heavier vehicle with a slower actuator (Case II), a lighter vehicle with a faster actuator (Case III), and a heavier vehicle with a faster actuator (Case IV).

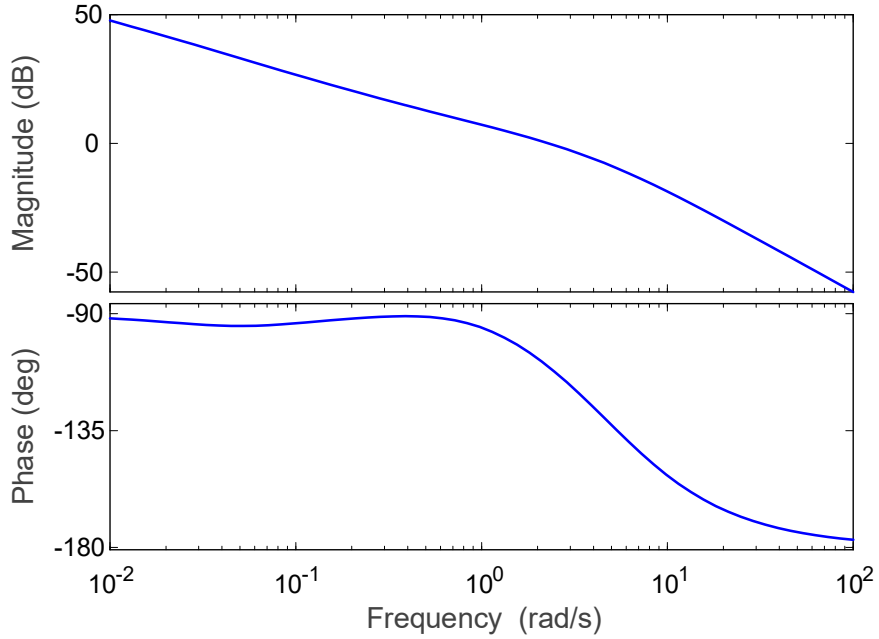


Figure 8. Open-loop Bode diagram of the h-CCODE-tuned PID-controlled cruise control system.

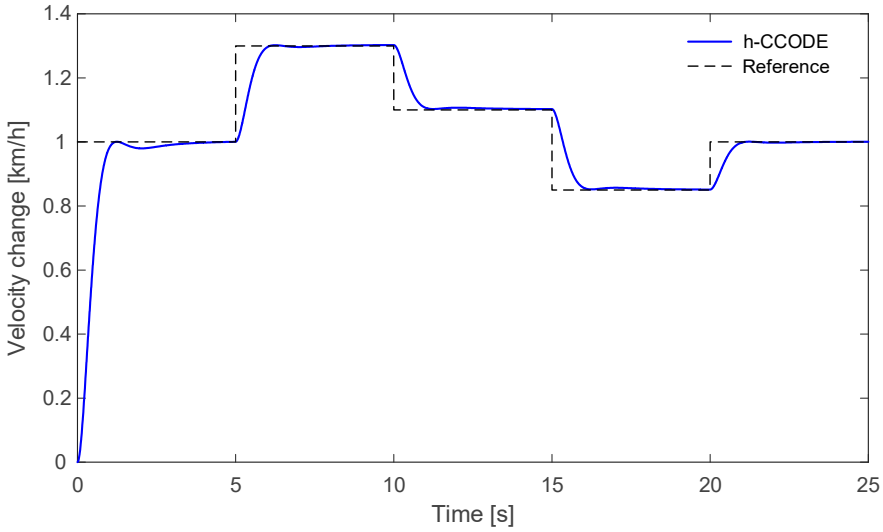


Figure 9. Reference tracking performance of the proposed controller under successive positive and negative speed variations.

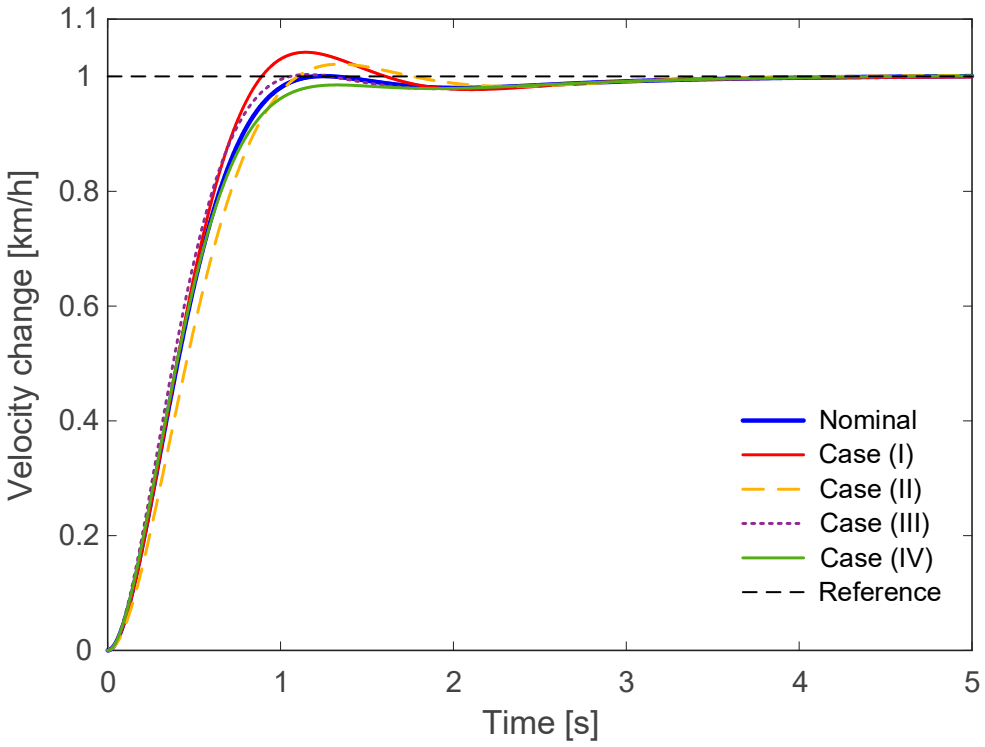


Figure 10. Closed-loop responses of the proposed controller under nominal and uncertain system parameters.

As shown in Figure 10, parameter variations influence the transient response of the system, particularly in terms of settling time and overshoot. However, all considered cases remain stable and continue to track the reference signal, demonstrating that the controller maintains acceptable closed-loop behavior under moderate parameter deviations. The quantitative performance under these uncertainty conditions is summarized in Table 9.

Table 9. Time-domain performance metrics under nominal and uncertain parameter conditions.

Cases	t_{rise} [s]	t_{set} [s]	p_{os} [%]
Nominal	0.6419	0.9968	0.0629
Case (I): $M = 1350$, $\tau = 0.24$	0.5764	2.3544	4.2151
Case (II): $M = 1600$, $\tau = 0.25$	0.6841	1.4251	2.1489
Case (III): $M = 1400$, $\tau = 0.19$	0.5965	2.1423	0.3294
Case (IV): $M = 1550$, $\tau = 0.18$	0.6737	2.1021	0.1097

From Table 9, it can be observed that the uncertainty cases affect the transient characteristics of the system in different ways. Although some cases yield shorter rise times than the nominal condition, this improvement is generally accompanied by increased settling time and/or overshoot. The nominal case, on the other hand, provides the most balanced overall response by combining low settling time with minimal overshoot. Despite the performance variations introduced by parameter uncertainty, the controller preserves stable responses in all considered cases, which supports the robustness of the proposed control approach.

6.6. Practical Relevance and Limitations

The results of this study are obtained by simulation, and their practical relevance should be assessed accordingly. The plant of Equation (10) is not a synthetic dataset but a physics-based model of longitudinal vehicle motion whose structure and parameters

originate from the automotive control literature and model-based design of this kind constitutes the usual first stage of controller development, preceding hardware-in-the-loop testing and in-vehicle calibration. Within this stage, several features of the present design bear directly on its implementability. The controller is a fixed-gain PID law, so the tuning is carried out entirely offline and the deployed controller requires no online optimization, learning, or adaptation, and can be executed on a standard electronic control unit. The gains lie within the practical bounds of Section 5.3. The design tolerates sensor noise and step-type road and wind loads (Section 6.4), retains a delay margin of about 0.53 s, and remains stable under simultaneous variations of mass, actuator gain, time constant, delay, and aerodynamic loading over ranges considerably wider than those of Table 9 (Section 6.5). Moreover, the tuning procedure makes no use of features specific to the benchmark model and can therefore be applied without modification to a model identified from a particular vehicle. As an additional simulation-based check beyond the model of Equation (10), the tuning framework was also examined in a microscopic traffic simulation of a congested urban intersection, in which traffic flow dynamics and interactions between vehicles are represented; h-CCODE yielded the highest proportion of vehicles traversing the network among the five algorithms compared (Supplementary Material, Section S2). The limitations of the study should nevertheless be stated explicitly. The controller has not been validated experimentally, the plant is linearized about a single operating point, the actuator saturation of Equation (7) is not included in the design model, and the disturbances are represented by deterministic step profiles rather than by recorded road and wind data. The conclusions drawn here therefore concern the relative merit of the tuning procedures and the robustness of the resulting design at the model level, and confirmation on an instrumented vehicle or a hardware-in-the-loop platform remains a necessary next step.

7. Conclusions

This study addressed the problem of optimal PID controller tuning for vehicle cruise control systems, where nonlinear dynamics and varying operating conditions make parameter selection challenging. To improve the search capability of the original centered collision optimizer (CCO), a hybrid approach integrating differential evolution operators was proposed, resulting in the h-CCODE algorithm. The performance of the proposed optimizer was first evaluated using the CEC2022 benchmark functions. The obtained results demonstrated that h-CCODE provides improved convergence behavior and more consistent solution quality compared to the original CCO, particularly for complex optimization landscapes. These findings indicate that the incorporation of DE operators enhances the balance between exploration and exploitation without degrading baseline performance. The effectiveness of the proposed approach was further validated through its application to a vehicle cruise control system. The h-CCODE-based PID controller achieved improved closed-loop performance, characterized by reduced settling time and lower overshoot while maintaining a comparable rise time. Comparative results with several optimization-based controllers from the literature confirmed that the proposed method offers a more favorable overall transient response. A controlled comparison with PSO, GWO, MGO, and QIO under identical tuning conditions further showed that h-CCODE attains the lowest mean tuning cost and the smallest dispersion over 30 independent runs, and that its performance is only weakly affected by the choice of the crossover rate. The disturbance rejection and robustness analyses further supported these findings. Under external disturbances and measurement noise, the proposed controller demonstrated competitive performance in terms of error-based indices. In addition, frequency-domain analysis indicated satisfactory stability margins, while tracking and parameter uncertainty tests showed that the controller maintains stable and well-behaved responses under varying conditions. Overall, the pro-

posed h-CCODE algorithm provides an effective framework for PID controller tuning in cruise control applications by improving convergence stability and ensuring consistent control performance.

As future work, the proposed framework can be extended to electric vehicle (EV) systems, where speed regulation, energy efficiency, and robustness are critical. The application of h-CCODE to more detailed nonlinear EV models and real-time control implementations represents a promising direction for further investigation. The disturbance rejection assessment could likewise be extended with measured road-grade and wind profiles, together with reference trajectories derived from standard driving cycles, which would allow the performance of the tuned controller to be quantified under recorded operating conditions. In particular, experimental validation of the tuned controller, first on a hardware-in-the-loop platform and subsequently on an instrumented vehicle, is required to confirm the model-level findings reported here.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app162010140/s1, Table S1: Algorithm-specific parameter settings of the compared algorithms; Table S2: Statistical results of h-CCODE, CCO, PSO, GWO, MGO, and QIO on the CEC2022 test suite (30 independent runs; CCO results from Table 2 of the main text); Table S3: Statistical results of the cost function obtained by h-CCODE, CCO, PSO, GWO, MGO, and QIO for the vehicle cruise control tuning problem (30 independent runs; CCO results from Table 3 of the main text); Figure S1: Convergence curves of the cost function obtained by h-CCODE, PSO, GWO, MGO, and QIO for the vehicle cruise control tuning problem (run with the best final cost for each algorithm); Table S4: Number of iterations required to reach given cost levels on the vehicle cruise control tuning problem (run with the best final cost for each algorithm; CCO values obtained from Figure 3 of the main text); Table S5: Sensitivity of h-CCODE to the crossover rate *CR* on the vehicle cruise control tuning problem (30 independent runs); Figure S2: Success rate, expressed as the percentage of vehicles successfully leaving the traffic network, obtained by the compared algorithms over 200 iterations; Table S6: Number of vehicles traversing the simulated traffic network and resulting success rate. References [40–44] are cited in the Supplementary Materials.

Author Contributions: C.T.: Formal analysis, Software, Validation, Writing—original draft. S.E.: Methodology, Software, Validation, Writing—original draft. D.I.: Methodology, Writing—reviewing and editing, Software, Validation. E.A.: Validation, Writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets analyzed during the current study are available from the corresponding author on reasonable request.

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

Appendix A. Mathematical Formulation of CCO

Appendix A.1. Head-On Collision Mechanics

Appendix A.1.1. Conservation of Momentum

For a closed system without external forces, the total momentum remains unchanged before and after collision:

$$\vec{p} = \vec{p}' \tag{A1}$$

For a single particle, momentum is defined as:

$$\vec{p} = m\vec{v} \tag{A2}$$

Accordingly, for a two-particle head-on collision,

$$\begin{aligned} \vec{p} &= \vec{p}_1 + \vec{p}_2 = m_1\vec{u}_1 + m_2\vec{u}_2 \\ \vec{p}' &= \vec{p}_1' + \vec{p}_2' = m_1\vec{v}_1 + m_2\vec{v}_2 \\ m_1\vec{u}_1 + m_2\vec{u}_2 &= m_1\vec{v}_1 + m_2\vec{v}_2 \end{aligned} \tag{A3}$$

where m_1 and m_2 denote the particle masses, $\vec{u}_1$ and $\vec{u}_2$ are the pre-collision velocities, and $\vec{v}_1$ and $\vec{v}_2$ are the post-collision velocities [35].

Appendix A.1.2. Perfectly Elastic Head-On Collision

In a perfectly elastic collision, both momentum and kinetic energy are conserved:

$$E_k = E_k' \tag{A4}$$

The kinetic energy of a single particle is

$$E_k = \frac{1}{2}m\left\|\vec{v}\right\|^2 \tag{A5}$$

Thus, for two colliding particles,

$$\frac{1}{2}m_1\left\|\vec{u}_1\right\|^2 + \frac{1}{2}m_2\left\|\vec{u}_2\right\|^2 = \frac{1}{2}m_1\left\|\vec{v}_1\right\|^2 + \frac{1}{2}m_2\left\|\vec{v}_2\right\|^2 \tag{A6}$$

By solving Equations (A3) and (A6), the post-collision velocities are obtained as:

$$\begin{aligned} \vec{v}_1 &= \frac{(m_1-m_2)\vec{u}_1+2m_2\vec{u}_2}{m_1+m_2} \\ \vec{v}_2 &= \frac{2m_1\vec{u}_1+(m_2-m_1)\vec{u}_2}{m_1+m_2} \end{aligned} \tag{A7}$$

Appendix A.1.3. Perfectly Inelastic Head-On Collision

For a perfectly inelastic collision, the two particles move together after impact, and therefore

$$\vec{v}_1 = \vec{v}_2 \tag{A8}$$

Combining Equation (A8) with the momentum conservation law yields

$$\vec{v}_1 = \vec{v}_2 = \frac{m_1\vec{u}_1 + m_2\vec{u}_2}{m_1 + m_2} \tag{A9}$$

Appendix A.1.4. Partially Inelastic Head-On Collision

In a partially inelastic collision, momentum is conserved while a portion of the kinetic energy is dissipated. The coefficient of restitution is defined as:

$$e = \frac{\left\|\vec{v}_2 - \vec{v}_1\right\|}{\left\|\vec{u}_1 - \vec{u}_2\right\|} \tag{A10}$$

By solving Equations (A3) and (A10), the post-collision velocities become

$$\begin{aligned}\vec{v}_1 &= \frac{(m_1-em_2)\vec{u}_1+(1+e)m_2\vec{u}_2}{m_1+m_2}\\ \vec{v}_2 &= \frac{(m_2-em_1)\vec{u}_2+(1+e)m_1\vec{u}_1}{m_1+m_2}\end{aligned} \tag{A11}$$

It is worth noting that Equation (A11) reduces to Equation (A7) when $e = 1$ and to Equation (A9) when $e = 0$.

Appendix A.1.5. Perfectly Elastic Head-On Collision in a Uniformly Moving Frame

If the collision is observed in a uniformly moving reference frame with velocity $\vec{\omega}$, the relative velocity is written as:

$$\vec{v}_\omega = \vec{v} - \vec{\omega} \tag{A12}$$

In that frame, the elastic head-on collision model becomes

$$\begin{aligned}\vec{v}_{\omega 1} &= \frac{(m_1-m_2)\vec{u}_{\omega 1}+2m_2\vec{u}_{\omega 2}}{m_1+m_2}\\ \vec{v}_{\omega 2} &= \frac{(m_2-m_1)\vec{u}_{\omega 2}+2m_1\vec{u}_{\omega 1}}{m_1+m_2}\end{aligned} \tag{A13}$$

Transforming these expressions back to the stationary frame gives

$$\begin{aligned}\vec{v}_1 = \vec{\omega} + \vec{v}_{\omega 1} = \vec{\omega} + \frac{(m_1-m_2)}{m_1+m_2}\left(\vec{\omega}-\vec{u}_1\right) - \frac{2m_2}{m_1+m_2}\left(\vec{\omega}-\vec{u}_2\right)\\ \vec{v}_2 = \vec{\omega} + \vec{v}_{\omega 2} = \vec{\omega} + \frac{(m_1-m_2)}{m_1+m_2}\left(\vec{\omega}-\vec{u}_2\right) - \frac{2m_1}{m_1+m_2}\left(\vec{\omega}-\vec{u}_1\right)\end{aligned} \tag{A14}$$

Although Equation (A14) can be simplified into the same result as Equation (A7), its transformed representation is retained because it provides a more convenient interpretation for the position update process in the optimizer.

Appendix A.2. Algorithmic Model of CCO

Appendix A.2.1. Initialization Phase

Like most metaheuristics, CCO starts by generating a random population within the search bounds:

$$\vec{x}_i = \vec{x}_l + \vec{rand}\left(\vec{x}_u - \vec{x}_l\right), \quad i = 1, 2, \ldots, P \tag{A15}$$

where P is the population size, $\vec{x}_l$ and $\vec{x}_u$ denote the lower and upper bounds, respectively, and $\vec{rand}$ is a random vector uniformly distributed in $[0, 1]$. The entire population can be represented by

$$X = \begin{bmatrix}\vec{x}_1\\ \vec{x}_2\\ \vdots\\ \vec{x}_i\\ \vdots\\ \vec{x}_P\end{bmatrix} = \begin{bmatrix}\vec{x}_{1,1} & \vec{x}_{1,2} & \cdots & \vec{x}_{1,j} & \cdots & \vec{x}_{1,Dim}\\ \vec{x}_{2,1} & \vec{x}_{2,2} & \cdots & \vec{x}_{2,j} & \cdots & \vec{x}_{2,Dim}\\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\ \vec{x}_{i,1} & \vec{x}_{i,2} & \cdots & \vec{x}_{i,j} & \cdots & \vec{x}_{i,Dim}\\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\ \vec{x}_{P,1} & \vec{x}_{P,2} & \cdots & \vec{x}_{P,j} & \cdots & \vec{x}_{P,Dim}\end{bmatrix} \tag{A16}$$

where Dim is the problem dimension.

Appendix A.2.2. Centered Collision Strategy

In CCO, the particle velocities in the collision equations are reinterpreted as solution positions. Accordingly, population members are updated through collision-inspired in-

teractions. By rewriting Equation (A14), the update rule associated with the transformed elastic collision can be expressed as:

$$\vec{x}_i^{t+1} = \vec{x}_j^t + \alpha(\vec{x}_j^t - \vec{x}_i^t) + (-1-\alpha)(\vec{x}_j^t - \vec{x}_i^t), \quad \alpha = \frac{(m_2 - m_1)}{m_1 + m_2} \tag{A17}$$

Similarly, the perfectly inelastic collision model in Equation (A9) can be rewritten as

$$\vec{x}_i^{t+1} = \beta\vec{x}_j^t + \vec{x}_i^t(1-\beta), \quad \beta = \frac{m_1}{m_1 + m_2} \tag{A18}$$

To combine these two mechanisms, a dimension-wise update operator is introduced:

$$\vec{x}_i^{t+1} = \vec{R}\vec{x}_j^t + (1-\vec{R})[\alpha(\vec{x}_j^t - \vec{x}_r^t) + (1-\alpha)(\vec{x}_j^t - \vec{x}_r^t) + \beta\vec{x}_j^t + \vec{x}_i^t(1-\beta)] \tag{A19}$$

where $\vec{x}_r^t$ is a randomly selected individual and $\vec{x}_j^t$ is a fitter individual than $\vec{x}_i^t$. The binary vector $\vec{R}$ determines which collision type is applied in each dimension:

$$\vec{R} = \begin{cases} 1, & \vec{rand} > 0 \\ 0, & otherwise \end{cases} \tag{A20}$$

To improve stochasticity and search diversity, the final centered collision strategy is defined as:

$$\vec{x}_i^{t+1} = \vec{R}\vec{x}_{j1}^t + (1-\vec{R})[\alpha(\vec{x}_{j2}^t - \vec{x}_{r1}^t) + (1-\alpha)(\vec{x}_{j3}^t - \vec{x}_{r2}^t) + \beta\vec{x}_{j4}^t + \vec{x}_i^t(1-\beta)] \tag{A21}$$

where $r1$ and $r2$ denote randomly chosen individuals, whereas $j1$, $j2$, $j3$, $j4$ are fitter individuals selected relative to $\vec{x}_i^t$. The control parameter α is adaptively adjusted as:

$$\alpha = \frac{1}{3}\left(\cos\left(\frac{t\pi}{T_{max}}\right) + 1\right) + \frac{1}{5} \tag{A22}$$

Equation (A21) serves as the unique position update operator of CCO and is referred to as the centered collision strategy.

Appendix A.2.3. Decorrelated Space

Performing the centered collision strategy only in the original space may reduce convergence speed and increase the tendency to be trapped in local optima. To alleviate this issue, CCO introduces a decorrelated space (DS), generated through an orthogonal transformation obtained from the covariance structure of a subset of individuals.

First, a data matrix $X_n \in \mathbb{R}^{n \times Dim}$ is formed using the $\frac{P}{2}$ worst-performing individuals. After column-wise centering, the zero-mean matrix is written as:

$$X_c = X_n - 1\vec{\mu} \tag{A23}$$

where $\vec{\mu} = \frac{1}{n}\sum_{i=1}^{n}\vec{x}_i$ is the mean vector. The covariance matrix is then computed by

$$C = \frac{1}{n}X_c^T X_c \tag{A24}$$

Next, eigenvalue decomposition is performed as:

$$C\vec{v}_i = \lambda_i\vec{v}_i, i = 1, 2, \ldots, Dim \tag{A25}$$

and thus

$$C = V\Lambda V^T \tag{A26}$$

where V contains the eigenvectors and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{Dim})$. The population is mapped into the decorrelated space through

$$X_{DS} = XV \tag{A27}$$

The centered collision strategy is then executed in DS as

$$\vec{x}_{i,DS}^{t+1} = \vec{R}\vec{x}_{j1,DS}^{t} + (1-\vec{R})[\alpha(\vec{x}_{j2,DS}^{t} - \vec{x}_{r1,DS}^{t}) + (1-\alpha)(\vec{x}_{j3,DS}^{t} - \vec{x}_{r2,DS}^{t}) + \beta\vec{x}_{j4,DS}^{t} + \vec{x}_{i,DS}^{t}(1-\beta)] \tag{A28}$$

Finally, the updated DS solution is projected back into the original space by

$$\vec{x}_{i}^{t+1} = \vec{x}_{i,DS}^{t+1}V^T \tag{A29}$$

This transformation-based update process enhances search diversity and helps the optimizer escape from local minima more effectively.

Appendix A.2.4. Space Allocation Strategy

Because the relative effectiveness of *OS* and *DS* may vary during the search, CCO employs an adaptive space allocation strategy. A greedy selection mechanism is first applied in each space. For the original space,

$$\begin{aligned} &k_{OS}^{t+1} = 1, \\ &\textbf{\textit{if}}\ f(\vec{x}_{i}^{t+1}) < f(\vec{x}_{i}^{t}), \\ &\quad \vec{x}_{i}^{t+1} = \vec{x}_{i}^{t+1}, \\ &\quad k_{OS}^{t+1} = k_{OS}^{t+1} + 1, \\ &\quad \textbf{\textit{else}} \\ &\quad \vec{x}_{i}^{t+1} = \vec{x}_{i}^{t} \end{aligned} \tag{A30}$$

Similarly, for the decorrelated space,

$$\begin{aligned} &k_{DS}^{t+1} = 1, \\ &\textbf{\textit{if}}\ f(\vec{x}_{i}^{t+1}) < f(\vec{x}_{i}^{t}), \\ &\quad \vec{x}_{i}^{t+1} = \vec{x}_{i}^{t+1}, \\ &\quad k_{DS}^{t+1} = k_{DS}^{t+1} + 1, \\ &\quad \textbf{\textit{else}} \\ &\quad \vec{x}_{i}^{t+1} = \vec{x}_{i}^{t} \end{aligned} \tag{A31}$$

Using the successful update counts of both spaces, the normalized allocation ratios are updated as:

$$\begin{aligned} kp_{OS}^{1} = kp_{DS}^{1} = 0.5, \\ kp_{OS}^{t+1} = \frac{k_{OS}^{t+1}/kp_{OS}^{t}}{k_{OS}^{t+1}/kp_{OS}^{t} + k_{DS}^{t+1}/kp_{DS}^{t}} \\ kp_{DS}^{t+1} = \frac{kp_{DS}^{t+1}/kp_{DS}^{t}}{kp_{OS}^{t+1}/kp_{OS}^{t} + kp_{OS}^{t+1}/kp_{OS}^{t+1}} = 1 - kp_{OS}^{t+1}, \\ 0.3 \le kp_{DS}^{t},\ kp_{OS}^{t} \le 0.7, \end{aligned} \tag{A32}$$

where kp_{OS}^{t} and kp_{DS}^{t} denote the proportions of individuals evolved in *OS* and *DS* at iteration t, respectively. Initially, both spaces are assigned equal shares, and thereafter

the ratios are adaptively adjusted while ensuring that each space receives at least 30% of the population.

References

1. Frank, A.A.; Liu, S.J.; Liang, S.C. Longitudinal Control Concepts for Automated Automobiles and Trucks Operating on a Cooperative Highway. *SAE Trans.* **1989**, *98*, 1308–1315.
2. Lewis, P.H.; Houghton, Y. *Basic Control Systems Engineering*; Prentice Hall: Upper Saddle River, NJ, USA, 1997.
3. Idir, A.; Zemmit, A.; Khettab, K.; Nesri, M.; Guedida, S.; Canale, L. Performance Analysis of Fractionalized Order PID Controller-Based on Metaheuristic Optimisation Algorithms for Vehicle Cruise Control Systems. *Sci. Eng. Technol.* **2024**, *5*, 1–14. [CrossRef]
4. Choi, K.-H.; Chang, H.-J. Improvement of Adaptive Cruise Control System Performance on Sloped Roads Based on Adaptive Neuro-Fuzzy Inference System. *IEEE Access* **2025**, *13*, 60519–60531. [CrossRef]
5. Zhao, X.; Zhao, C.; Li, K.; Hu, Y.; Ren, H.; Luo, Y. Functional Modification Based on Insufficiency Identification and Fault-Tolerant Control for Cooperative Adaptive Cruise Control Systems. *J. Intell. Transp. Syst.* **2025**, 1–19. [CrossRef]
6. Zhang, Y.; Lin, Y.; Qin, Y.; Dong, M.; Gao, L.; Hashemi, E. A New Adaptive Cruise Control Considering Crash Avoidance for Intelligent Vehicle. *IEEE Trans. Ind. Electron.* **2024**, *71*, 688–696. [CrossRef]
7. Bennett, S. Development of the PID Controller. *IEEE Control Syst.* **1993**, *13*, 58–62. [CrossRef]
8. Aström, K.J.; Murray, R.M. *Feedback Systems*; Princeton University Press: Princeton, NJ, USA, 2008.
9. Meng, X.; Yu, H.; Zhang, J.; Yang, Q.; Fu, C. Adaptive Fault-Tolerant Cooperative Optimization Control for PMSM Servo System with Input Saturation and Multisource Disturbances. *IEEE Trans. Power Electron.* **2025**, *40*, 6506–6518. [CrossRef]
10. Tong, L.; Cui, D.; Wang, C.; Peng, L. A Novel Zero-Force Control Framework for Post-Stroke Rehabilitation Training Based on Fuzzy-PID Method. *Intell. Robot.* **2024**, *4*, 125–145. [CrossRef]
11. Viadero-Monasterio, F.; Meléndez-Useros, M.; Jiménez-Salas, M.; López Boada, B.; Jesús López Boada, M. Key Influencing Factors in Vehicle Platoons: A Systematic Study and Review. *Evol. Syst.* **2025**, *16*, 116. [CrossRef]
12. Viadero-Monasterio, F.; Meléndez-Useros, M.; Boada, B.L.; Boada, M.J.L. Static Output-Feedback LQR Control for Predecessor-Leader Following Vehicle Platooning With Energy Efficiency Analysis via Aerodynamic Drag Reduction. *IEEE Open J. Intell. Transp. Syst.* **2026**, *7*, 1329–1340. [CrossRef]
13. Murugan, N.; Mohamed Ismail, M.R. An Efficient Coordinated Observer LQR Control in a Platoon of Vehicles for Faster Settling Under Disturbances. *World Electr. Veh. J.* **2025**, *16*, 28. [CrossRef]
14. Ziegler, J.G.; Nichols, N.B. Optimum Settings for Automatic Controllers. *J. Fluids Eng.* **1942**, *64*, 759–765. [CrossRef]
15. Cohen, G.H.; Coon, G.A. Theoretical Consideration of Retarded Control. *J. Fluids Eng.* **1953**, *75*, 827–834. [CrossRef]
16. Carbas, S.; Toktas, A.; Ustun, D. Introduction and Overview: Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications. In *Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications*; Springer: Singapore, 2021; pp. 1–9.
17. Prasanna, V.; Nelson, A.; Hnaumanthakari, S.; Kumar, V.K.; Kirubakaran, S.; Kumar, M.J. Metaheuristic Algorithm for Automatic Cruise Control System. In *Proceedings of the 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 20–22 October 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 206–211.
18. Rout, M.K.; Sain, D.; Swain, S.K.; Mishra, S.K. PID Controller Design for Cruise Control System Using Genetic Algorithm. In *Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 4170–4174.
19. Pradhan, R.; Majhi, S.K.; Pradhan, J.K.; Pati, B.B. Antlion Optimizer Tuned PID Controller Based on Bode Ideal Transfer Function for Automobile Cruise Control System. *J. Ind. Inf. Integr.* **2018**, *9*, 45–52. [CrossRef]
20. Izci, D.; Ekinci, S. A Novel Hybrid ASO-NM Algorithm and Its Application to Automobile Cruise Control System. In *2nd International Conference on Artificial Intelligence: Advances and Applications*; Mathur, G., Bundele, M., Mahendra, L., Paprzycki, M., Eds.; Springer: Singapore, 2022; pp. 333–343.
21. Ekinci, S.; Izci, D. Enhanced Reptile Search Algorithm with Lévy Flight for Vehicle Cruise Control System Design. *Evol. Intell.* **2023**, *16*, 1339–1351. [CrossRef]
22. Heybetli, F.; Danayiyen, Y.; Taşdemir, A.B.; Şenyiğit, Ş. Comparative Analysis of Metaheuristic Algorithms in PID-Based Vehicle Cruise Control Systems. *Electrica* **2025**, *25*, 1–16. [CrossRef]
23. Izci, D.; Ekinci, S.; Kayri, M.; Eker, E. 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.* **2022**, *13*, 453–468. [CrossRef]
24. Izci, D.; Ekinci, S. Comparative Performance Analysis of Slime Mould Algorithm for Efficient Design of Proportional–Integral–Derivative Controller. *Electrica* **2021**, *21*, 151–159. [CrossRef]
25. Yang, X.-S. Nature-Inspired Optimization Algorithms: Challenges and Open Problems. *J. Comput. Sci.* **2020**, *46*, 101104. [CrossRef]
26. Mirjalili, S. *Evolutionary Algorithms and Neural Networks*; Springer International Publishing: Cham, Switzerland, 2019; Volume 780.

27. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 10031–10061. [CrossRef]
28. Ekinci, S.; Turkeri, C.; Izci, D.; Abualigah, L.; Bajaj, M.; Blazek, V. Optimizing Steam Condenser Efficiency: Integrating Logarithmic Spiral Search and Greedy Selection Mechanisms in Gazelle Optimizer for PI Controller Tuning. *Results Eng.* **2024**, *24*, 103501. [CrossRef]
29. Biswas, S.; Singh, G.; Maiti, B.; Ezugwu, A.E.-S.; Saleem, K.; Smerat, A.; Abualigah, L.; Bera, U.K. Integrating Differential Evolution into Gazelle Optimization for Advanced Global Optimization and Engineering Applications. *Comput. Methods Appl. Mech. Eng.* **2025**, *434*, 117588. [CrossRef]
30. Biswas, S.; Maiti, B.; Singh, G.; Ezugwu, A.E.; Saleem, K.; Abualigah, L.; Smerat, A.; Bera, U.K. A Novel Hybrid Optimizer Based on Coati Optimization Algorithm and Differential Evolution for Global Optimization and Constrained Engineering Problems. *Int. J. Comput. Intell. Syst.* **2025**, *18*, 157. [CrossRef]
31. Maiti, B.; Biswas, S.; Ezugwu, A.E.-S.; Bera, U.K.; Alzahrani, A.I.; Alblehai, F.; Abualigah, L. Enhanced Crayfish Optimization Algorithm with Differential Evolution's Mutation and Crossover Strategies for Global Optimization and Engineering Applications. *Artif. Intell. Rev.* **2025**, *58*, 69. [CrossRef]
32. Ekinci, S.; Turkeri, C.; Gokalp, I.; Li, Y.; Li, D. High-Performance Temperature Regulation of Nonlinear CSTRs via a Hybrid Stellar Oscillation Optimizer and Differential Evolution-Based PID-F Control. *Sci. Rep.* **2026**, *16*, 7713. [CrossRef] [PubMed]
33. Ekinci, S.; Izci, D.; Kayri, M.; Elsayed, F.; Salman, M. A Novel Hybridization of Birds of Prey-Based Optimization with Differential Evolution Mutation and Crossover for Chaotic Dynamics Identification. *Sci. Rep.* **2025**, *15*, 43451. [CrossRef] [PubMed]
34. Das, S.; Suganthan, P.N. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [CrossRef]
35. Lang, Y.; Gao, Y.; Chen, T.; Wang, H. Centered Collision Optimizer: A Novel and Efficient Physics-Based Metaheuristic Optimization Algorithm for Solving Complex Real-World Engineering Optimization Problems. *Comput. Methods Appl. Mech. Eng.* **2026**, *448*, 118491. [CrossRef]
36. Kuyu, Y.Ç.; Vatansever, F. Advanced Metaheuristic Algorithms on Solving Multimodal Functions: Experimental Analyses and Performance Evaluations. *Arch. Comput. Methods Eng.* **2021**, *28*, 4861–4873. [CrossRef]
37. Alorf, A. A Survey of Recently Developed Metaheuristics and Their Comparative Analysis. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105622. [CrossRef]
38. Turkeri, C.; Ekinci, S.; Yüksek, G.; Li, D. A Modified Enzyme Action Optimizer-Based FOPID Controller for Temperature Regulation of a Nonlinear Continuous Stirred Tank Reactor. *Fractal Fract.* **2025**, *9*, 811. [CrossRef]
39. Kumar, A.; Price, K.V.; Mohamed, A.W.; Hadi, A.A.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2022 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization*; Technical Report; Nanyang Technological University: Singapore, 2021.
40. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In *Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995*; IEEE: Piscataway, NJ, USA, 1995; pp. 1942–1948.
41. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
42. Abdollahzadeh, B.; Gharehchopogh, F.S.; Khodadadi, N.; Mirjalili, S. Mountain Gazelle Optimizer: A New Nature-Inspired Metaheuristic Algorithm for Global Optimization Problems. *Adv. Eng. Softw.* **2022**, *174*, 103282. [CrossRef]
43. Zhao, W.; Wang, L.; Zhang, Z.; Mirjalili, S.; Khodadadi, N.; Ge, Q. Quadratic Interpolation Optimization (QIO): A New Optimization Algorithm Based on Generalized Quadratic Interpolation and Its Applications to Real-World Engineering Problems. *Comput. Methods Appl. Mech. Eng.* **2023**, *417*, 116446. [CrossRef]
44. Ekinci, S.; Izci, D.; Abualigah, L.; Zitar, R.A. A Modified Oppositional Chaotic Local Search Strategy Based Aquila Optimizer to Design an Effective Controller for Vehicle Cruise Control System. *J. Bionic Eng.* **2023**, *20*, 1828–1851. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.