Indian Journal of Pure & Applied Physics
Vol. 64, August 2026, pp. 894-914
DOI: 10.56042/ijpap.v64i8.31054

Optimal Robust Design of a Complex Fractional-Order PID Controller for Automotive Cruise Control Applications

Biresh Kumar Dakua[a*], Sonalika Mishra[a] & Sarada Prasanna Sahoo[b]

[a]Department of Electrical Engineering, Silicon Institute of Technology, Sambalpur 768 200, India

[b]Department of Electrical & Electronics Engineering, Vedang Institute of Technology, Khurda 752 010, India

Received: 2nd May 2026; accepted: 7th July 2026

Automobile Cruise Control (ACC) systems regulate vehicle speed at a predefined setpoint, thereby reducing driver workload, enhancing driving comfort, and improving fuel efficiency. This paper presents a control-oriented investigation of ACC speed regulation using integer-order (IO) and non-integer-order (NIO) controllers, including proportional integral derivative (PID), tilt integral derivative (TID), fractional-order PID (FOPID), fractional-order TID (FOTID), and complex fractional-order PID (COPID) controllers. The analysis, design, and tuning of the seven-parameter COPID controller constitute the primary contribution of this work. A time-domain-based parameter estimation framework is developed to ensure satisfactory control performance. Controller parameters are optimized using four population-based metaheuristic algorithms: Artificial Rabbits Optimization (ARO), Arithmetic Optimization Algorithm (AOA), Supply-Demand Optimization (SDO), and Biogeography-Based Optimization (BBO). Error-based and performance-based objective functions are incorporated for comprehensive tuning of IO and NIO controllers. The effectiveness, robustness, and consistency of the optimization techniques are systematically evaluated through statistical analysis. Simulation results demonstrate that the BBO-optimized, ZLG-tuned COPID controller consistently outperforms the other controllers, achieving reduced rise time (T_R), settling time (T_S), and percentage overshoot (M_P). Robustness analysis under parameter variations and external disturbances further confirms superior disturbance rejection and stable performance, establishing COPID as a reliable and efficient control strategy for ACC systems.

Keywords: Automobile cruise control, Complex-order PID controller, Fractional-order controller, Metaheuristic optimization, Robust control

1 Introduction

A review of the existing literature reveals that most existing commercial ACC systems employ PID controllers due to their simplicity and effective performance. Successful implementation of different metaheuristic algorithm based PID[1,2], an Ant lion optimizer (ALO) based PID[3,4], an AOA based PID[5], a Reptile search optimiser (RSO) based PID[6], an Aquila optimizer (AO) based PID[7], a Pattern search (PS) algorithm based PID[8], a Genetic algorithm (GA) based PID[9], a Grey wolf optimizer (GWO) based PID[10], a hybrid Atom search algorithm (ASO) and Nelder-Mead (NM) optimisation based PID[11,12], a modified BAT algorithm based PIDA[13], and an analytical evaluation of PID for cruise control[14] applications are provided in literature.

However, PID controllers suffer from limitations when dealing with nonlinear dynamics, parameter uncertainties, and varying operating conditions commonly present in automobile systems. The last few decades have seen the rise of fractional-order PID controllers, which extend traditional PID to non-integer orders. They are recognized for delivering superior closed-loop performance and robustness compared to classical controllers. The fractional-order control (FOC) has various advantages over the IOPID controller[15,16]. Very scant literature employs the FOPID controller for addressing the ACC problem. The supremacy of the FOPID controller over the IOPID controller employed for a cruise control system is well explained[17]. Enhanced control performance has been demonstrated using the Artificial Hummingbird Algorithm (AHA) and FOPID cruise controller[18]. A meta-heuristic algorithm-based FOPID controller is proposed for the cruise control system[19].

The TID controllers have emerged as an extension of classical PID controllers. The additional tilted component in this controller expands the stability region[20]. The TID controller has shown enhanced performance for diverse applications[21], but the use of

*Corresponding author: E-mail: bireshdakua@gmail.com

TID as a cruise controller is rare in the literature. Likewise, the fractional modification of the TID is the FOTID controller[22]. Even though it is well proven that FOTID produces excellent robustness to parameter variations in different engineering applications, the use of FOTID as an automobile cruise controller is not mentioned in any literature.

Building upon these developments, the COPID controller integrates fractional calculus with complex order dynamics to further enhance control flexibility and performance. The successful implementations of the COPID controller exist for various control problems. A COPID controller is formulated for a first-order plus time delay (FOPTD) system[23], where it has shown improved performance. Complex-order PI-PD Controllers for Process Control applications have been successfully established[24]. The worth application of complex-order PI/PID controllers is justified[25,26]. COPID controller design for improved blood-glucose regulation in Type-I diabetes patients is demonstrated[27]. Similar use of COPID towards a fractional-order DC motor system and a buck converter system is presented[28,29].

It is evident from the above-mentioned studies that for the best control action, optimal tuning of the controllers is essential, which has been a significant challenge. However, the dominance of non-deterministic methods like metaheuristic algorithms over deterministic approaches has led to an increased application of metaheuristic algorithms for optimal tuning of controllers. Despite the existing methodologies and their many applications, it is common to encounter newly developed metaheuristics. The free lunch theorem[30], which is the motivation behind this, asserts that all stochastic-based algorithms will ultimately exhibit comparable performance on average when all available optimization problems are taken into account. This situation arises because certain algorithms excel in addressing specific problems while being relatively ineffective for others.

In the studies, different types of objective functions are examined by researchers following the optimization algorithms. The error-based objective functions used are the integral of the square value of error (ISE), the integral of the absolute value of error (IAE), the integral of the time-weighted absolute value of error (ITAE), and the integral of the time-weighted square value of error (ITSE). Furthermore, user-defined objective functions that incorporate a mix of metrics such as percentage overshoot, settling time, peak time, rise time, steady-state error, phase margin, and gain margin are also applied. A popular user-defined fitness function (ZLG)[31] that includes percentage overshoot, settling time, rise time, and steady-state error.

The review of the existing literature reveals that the design and analysis of the COPID controller for cruise control systems has not yet been investigated. In the majority of earlier studies on ACC, researchers have employed error-based objective functions like ISE, IAE, ITSE, and ITAE and a performance-based objective function, ZLG, to improve system response characteristics. Motivated by this limitation, along with the existing performance indices, this article also validates the applicability of different time-square-based error functions like IT^2SE, and IT^2AE . These objective functions are used for the optimal tuning of COPID, FOTID, TID, and PID controllers for speed regulation in an ACC system. A wide variety of optimization algorithms have been developed[32], inspired by domains such as swarm intelligence, economics, mathematics, and biogeography. However, comparative studies across these distinct categories remain limited. To address this gap, the present study selects four representative algorithms from different domains: ARO[33], a swarm intelligence-based algorithm; SDO[34], inspired by economic principles; the AOA[35], based on mathematical concepts; and BBO[36], inspired by biogeographical processes. A balanced evaluation of the efficacy of both conventional and newly created metaheuristics for fine-tuning of integer and NIO controllers in cruise control applications is made possible by this selection.

2 System Under Study

2.1 Mathematical Modelling of Automotive Cruise Control (ACC) System

Consider an automobile system shown in Fig. 1 demonstrating the different forces acting upon it. The longitudinal dynamics of the vehicle, captured considering the force balance principle[1-4], are presented below. The engine driving force (F_d) acting on the automobile can be expressed as;

$$F_d = F_m + F_a + F_g. \quad \text{... (1)}$$

Mathematically, the inertial force can be presented as $F_m = M\frac{d\mathcal{V}}{dt}$, the aerodynamic drag force $F_a = C_a(\mathcal{V} - \mathcal{V}_w)^2$, and the downgrade force $F_g = Mg\sin\theta$. Further, $M, C_a, \mathcal{V}, \mathcal{V}_w, \theta,$ and u represent the

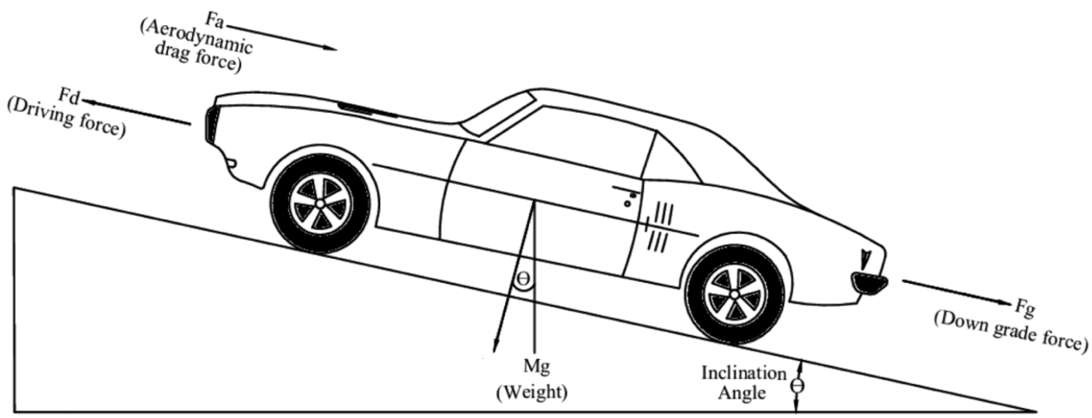

Fig. 1 — An Automobile System

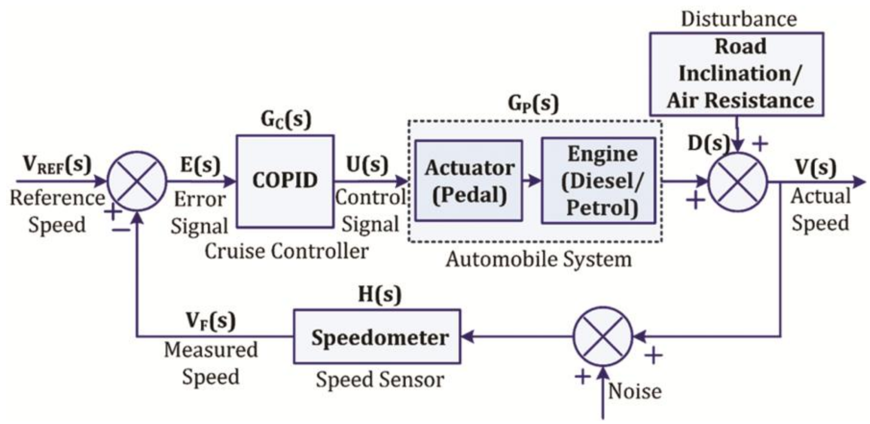

Fig. 2 — A closed-loop ACC System

automobile mass including driver, the aerodynamic drag coefficient, the vehicle speed, the wind gust speed, the road inclination angle, and the engine throttle angle, respectively. In practice, the driving force $F_d = \frac{C_1 e^{\tau s}}{1+sT}$ is usually presented considering a first-order plus time delay (FOPTD) model. Here, τ represents the reaction time of the driver.

As Eq. (1) describes a nonlinear dynamical model, some linearization in the above model can be done assuming zero initial values and zero input disturbances, i.e. $\theta = 0$, and $\mathcal{V}_w = 0$. Now, considering $\mathcal{V}$ and F_d As the state variables, the mathematical representation of the automobile system is as follows;

The state equation

$$\mathcal{V}' = \frac{1}{M}(F_d - C_a \mathcal{V}^2),$$
$$F_d{}' = \frac{1}{T}(C_a u(t-T) - F_d) \quad \text{... (2)}$$

The output equation

$$y = \mathcal{V} \quad \text{... (3)}$$

As the above simplified model is still nonlinear, adopting the procedure[3,4], the linearization at an appropriate equilibrium point can be done. Further, following the procedure as mentioned in[3,4], an equivalent transfer function model can also be obtained from the above linearized automobile model as follows;

$$G_P(s) = \frac{\Delta V(s)}{\Delta U(s)} = \frac{\left(\frac{C_1}{\tau MT}\right)}{\left(s+\frac{2C_a\mathcal{V}}{M}\right)\left(s+\frac{1}{T}\right)\left(s+\frac{1}{\tau}\right)} \quad \text{... (4)}$$

By considering the parameters of the ACC system as mentioned in Table 1, the mathematical transfer function of the cruise control system can be presented as;

$$G_P(s) = \frac{\Delta V(s)}{\Delta U(s)} = \frac{2.4767}{s^3+6.0476s^2+5.2856s+0.2380} \quad \text{... (5)}$$

The primary objective of the ACC system, illustrated in Fig. 2, is to regulate and maintain the automobile speed at the desired reference speed (V_{REF}) specified by the driver. The applied cruise controller compares the reference speed (V_{REF}) with

Table 1 — Parametric values of the ACC system[3]

ACC Parameters	Values
Actuator Constant (C_1)	743
Aerodynamic Drag Coefficient (C_a)	1.19 N/(m/s)^2
Mass of the Vehicle with Driver (M)	1500 kg
Observation Time of Driver (T)	1s
Reaction time of Driver (τ)	0.2s
Vehicle Reference Speed (V_{REF})	30 km/hr

the measured vehicle speed (V_F). The resulting error signal is used to regulate the engine airflow by adjusting the throttle angle $U(s)$, thereby controlling the vehicle output speed $V(s)$. The frequency response study of the ACC system, as shown in Fig. 3, indicates a gain margin of $GM = 13.2$ dB at a frequency $\omega_{PC} = 0.0331$ rad/s and a phase margin $PM = 89^o$ at frequency $\omega_{GC} = 0.0068$ rad/s. The system also exhibits a delay margin $DM = 227$s, indicating a high tolerance to time delay. These stability margins confirm that the closed-loop system is stable and demonstrate adequate robustness against gain and phase variations.

The major control issue in this case study is the steady-state inaccuracy observed in the step response of the ACC system, as shown in Fig. 4. The system exhibits a rise time of $T_R = 2.7118$ s, a settling time of $T_S = 8.4390$ s, a percentage overshoot of $M_P = 7.328$ %, and a steady-state error of $E_{SS} = 0.0926$. Therefore, the primary control objective is to minimize the steady-state error, which is approximately 9.26 %, while maintaining acceptable transient response characteristics.

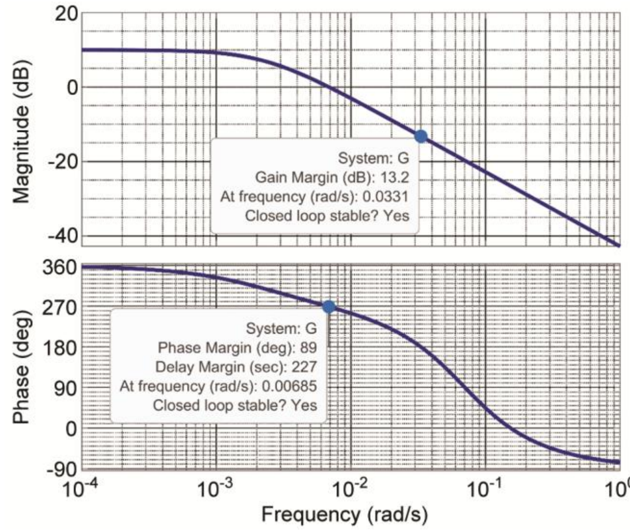

Fig. 3 — Frequency response of the uncompensated ACC system

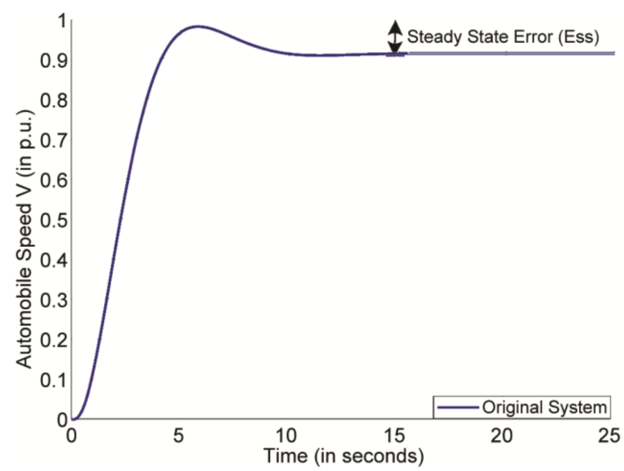

Fig. 4 — Step response of the uncompensated ACC system

2.2 Integer Order (IO) and Non-Integer Order (NIO) Controllers

In control systems, controllers are employed to compensate for inherent limitations and uncertainties in the plant model, thereby enabling the system to achieve the desired performance specifications. Various studies suggested the superiority of NIO controllers over their IO counterparts[15,36]. In this study, both IO and NIO controllers, as presented in Fig. 5, are designed, and their effectiveness is systematically evaluated and compared in achieving the desired control objectives.

2.2.1 Fractional Order PID (FOPID) Controller

A fractional modification of the PID controller is the FOPID ($PI^\lambda D^\mu$) controller whose transfer function can be presented as;

$$G_{C\_FOPID}(s) = K_P + \frac{K_I}{s^\lambda} + K_D s^\mu, \lambda, \mu > 0 \quad \text{... (6)}$$

Here, the proportional gain constant K_P magnifies the strength of the signal, the integral gain constant K_I eliminates the error at steady state, and the derivative gain K_D suppresses the overshoot. Further, λ and μ are fractional exponents that range within $(0-2)$.

The PID controller is a specific case of the FOPID, where $\lambda = 1$ and $\mu = 1$.

$$G_{C\_PID}(s) = K_P + \frac{K_I}{s} + K_D s, \lambda, \mu = 1 \quad \text{... (7)}$$

A five-parameter tuneable FOPID comprising fractional integral and derivative components usually provides better performance compared to the classical three-parameter tuneable PID controller.

2.2.2 Fractional Order TID (FOTID) Controller

A modified FOPID controller with a tilt compensation is classified as a FOTID ($TI^\lambda D^\mu$)

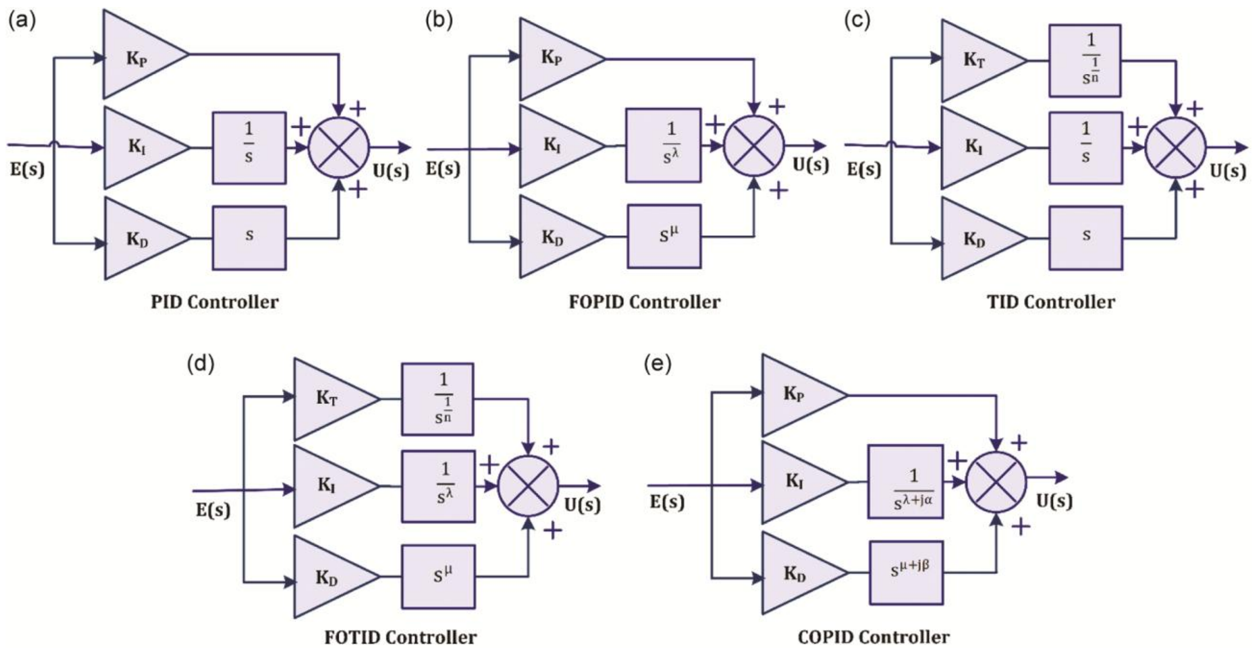

Fig. 5 — Various integer and non-integer order controllers

controller whose dynamics can be presented as follows;

$$G_{C\_FOTID}(s) = \frac{K_T}{s^{\frac{1}{n}}} + \frac{K_I}{s^{\lambda}} + K_D s^{\mu}, \lambda, \mu, \text{and } n > 0 \quad ... (8)$$

This controller comprises a proportional gain K_T, a tilt factor $s^{\frac{1}{n}}$ along with the fractional integral and derivative. The non-zero tilt exponent n usually ranges within $(1-10)$. The tilt factor helps to modify the system response and thereby can match the properties of Bode's ideal transfer function. The TID controller is a similar specific case of the FOTID controller where $\lambda = 1$, and $\mu = 1$. The six-parameter tunable $TI^{\lambda}D^{\mu}$ controller usually provides enhanced disturbance rejection and robustness to parameter variation characteristics.

2.2.3 Complex Order PID (COPID) Controller

A more generalized version of the NIO controller is a complex fractional-order PID (COPID) controller, where the real-valued fractional integral and derivatives are replaced by appropriate complex values. The dynamics of a COPID controller with a complex fractional-order integrator and differentiator can be presented mathematically as follows;

$$G_{C\_COPID}(s) = K_P + \frac{K_I}{s^{\lambda+j\alpha}} + K_D s^{\mu+j\beta}, \lambda, \mu > 0, \mathit{and}\ \alpha, \beta \in \mathcal{R} \quad ... (9)$$

Here, while the exponents λ, μ lie within$(0-2)$, the exponents α, β are within $(0-1)$ following the constraints of realizability. It can be observed here that the FOPID controller can be derived from the COPID, substituting $\alpha = 0$ and $\beta = 0$. The seven-parameter tuneable COPID controller provides enhanced degrees of freedom compared to the FOPID or FOTID controller.

2.3 Realization of Complex Fractional Order Elements

The physical realization of a fractional order integrator and differentiator is difficult due to infinite memory requirements. Therefore, approximations like the Oustaloup Recursive Approximation (ORA)[36], the Continuous Fractional Expansion (CFE), etc., are essential during the implementation of these NIO elements. The most common procedure is the ORA method that approximates the NIO integrators and differentiators in the s-domain over a specified frequency range $(\omega_{b,}\omega_h)$. Here, $\omega_{b,}$ and ω_h are the lower and the upper cut-off frequency, respectively.

Now, within the specified frequency range $\omega \in [\omega_b, \omega_h]$, the Oustaloup filter is expressed as[37];

$$G_F(s) = s^{\vartheta} = K \prod_{k=-N}^{N} \frac{s+\omega_Z}{s+\omega_P} \quad ... (10)$$

The zero ω_Z, the pole ω_P, and the gain K of the system can be calculated from the following equations.

$$\omega_Z = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+N+\frac{1}{2}(1-\beta)}{2N+1}},$$

$$\omega_P = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+N+\frac{1}{2}(1+\beta)}{2N+1}}, K = \omega_h^{\beta}, \text{ and } \omega_b.\omega_h = 1$$

Here, while ϑ represents the order of the NIO derivative ($\vartheta > 0$), N is the approximation order, and $2N + 1$ is the order of the Oustaloup filter. In this study, $\omega \in [10^{-3}, 10^{3}]$ Oustaloup's 11th-order filter approximation is considered for the analysis and simulation of NIO controllers in MATLAB[38].

Further, due to the presence of a complex order differentiator, the analysis of COPID is relatively difficult. The following procedure can be adopted for its corresponding fractional order conversion and relevant approximations.

The COPID controller, as expressed in Eq. (9), can be presented in realizable form as follows;

$$G_{C\_COPID}(s) = K_P + K_I\, \Re\{s^{-(\lambda+j\alpha)}\} + K_D\, \Re\{s^{\mu+j\beta}\} \quad \text{... (11)}$$

Now, using the real part approximation, the derivative term can be expressed as;

$$s^{\mu+j\beta} = s^{\mu}.s^{j\beta} = s^{\mu}.\exp^{\ln\{s^{j\beta}\}} = s^{\mu}.\exp^{j\beta.\ln(s)} =$$
$$s^{\mu}\,[\cos\{\beta.\ln(s)\} + j\sin\{\beta.\ln(s)\}],$$
$$\Rightarrow \Re\{s^{\mu+j\beta}\} = s^{\mu}\cos\{\beta.\ln(s)\} \quad \text{... (12)}$$

Likewise, following the real part approximation, the integrator term can be presented as;

$$\Re\{s^{-(\lambda+j\alpha)}\} = s^{-\lambda}\cos\left\{\alpha.\ln\left(\frac{1}{s}\right)\right\}$$
$$= s^{-\lambda}\cos\{\alpha.\ln(s^{-1})\}$$
$$= s^{-\lambda}\cos\{-\alpha.\ln(s)\}$$

As, $\cos\{-\theta\} = \cos\{\theta\}$; the above equation becomes;

$$\Rightarrow \Re\{s^{-(\lambda+j\alpha)}\} = s^{-\lambda}\cos\{\alpha.\ln(s)\} \quad \text{... (13)}$$

Now, as the complex-order terms are transformed to equivalent fractional order terms, the above OFA procedure can be adopted for the realization of complex fractional order integrals and derivatives. Fig. 6 illustrates the approximated model of the COPID controller employed for its practical realization and subsequent analysis and simulation in this study.

2.4 Parameter Tuning of Controllers

Proper tuning and precise estimation of controller parameters play a vital role in achieving the desired system performance. In practice, the tuning procedure of controllers is carried out following either a time domain or a frequency domain-based procedure. Usually, suitable error-based or performance-based objective functions are designed for the same and gets optimized adopting time or frequency domain methods. As error minimization and achieving optimal system performance are the needs of any control system, this article uses both error- and performance-based cost functions and considers different optimization techniques for the time domain-based parameter estimation of the applied controllers.

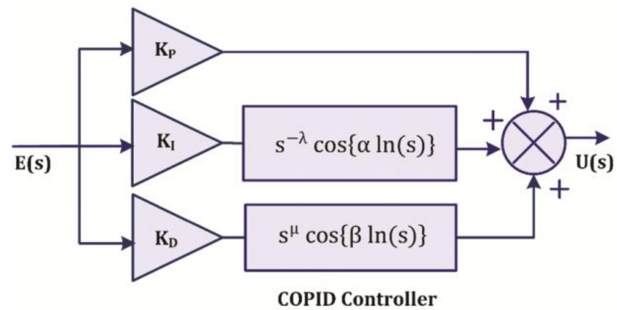

Fig. 6 — Approximation model of COPID controller

This article examines some commonly used error-based objective functions like ISE, IAE, ITAE, and ITSE, along with two more aggressive time-square error functions like IT^2AE and IT^2SE. Further, it also examines a performance-based cost function, i.e. ZLG as presented below;

i Error-based Cost function:

$$\min\ J = f\{t, e(t)\} \quad \text{... (14)}$$

- ISE: $J = \int_0^{t_{sim}} e^2(t)\,dt$
- IAE: $J = \int_0^{t_{sim}} |e(t)|\,dt$
- ITSE: $J = \int_0^{t_{sim}} te^2(t)\,dt$
- ITSE: $J = \int_0^{t_{sim}} te^2(t)\,dt$
- ITAE: $J = \int_0^{t_{sim}} t|e(t)|\,dt$
- IT^2SE: $J = \int_0^{t_{sim}} t^2e^2(t)\,dt$
- IT^2AE: $J = \int_0^{t_{sim}} t^2|e(t)|\,dt$

ii Performance-based Cost function:

$$\min\ J = f(T_R, T_S, M_P, E_{SS}) \quad \text{... (15)}$$

ZLG:

$$J = \{1 - \exp(-\rho)\}.(M_P + E_{SS}) + \exp(-\rho).(T_S - T_R)$$

Here, ρ denotes the weighting factor, whose typical values lie within the range [0.7, 1.5]. Moreover, (M_P) represents the peak overshoot, (E_{SS}) denotes the steady-state error, (T_R) is the rise time, and (T_S) is the settling time. It is important to note that for $\rho > 0.7$, the percentage overshoot and steady-state error are effectively suppressed, whereas for $\rho < 0.7$, a reduction in the rise time (T_R) and settling time (T_S) is observed. Based on the analysis conducted in this study, the most favourable system response is achieved for $\rho = 1$.

Further, the search region for the controller parameters during the optimization based tuning procedure is provided below;

PID: $3 \le K_P, K_I, K_D \le 4$
TID: $3 \le K_T, K_I, K_D \le 4$, and $1 \le n \le 10$
FOPID: $3 \le K_P, K_I, K_D \le 4$, and $0 \le \lambda, \mu \le 2$
FOTID: $3 \le K_T, K_I, K_D \le 4$, & $0 \le \lambda, \mu \le 2$, and $1 \le n \le 10$
COPID: $3 \le K_P, K_I, K_D \le 4$, & $0 \le \lambda, \mu \le 2$, and $0 \le \alpha, \beta \le 1$

The flow chart demonstrating the above tuning procedure is presented in Fig. 7. Different nature inspired optimization algorithms used for the parameter estimation procedure are discussed in the next section.

3 Optimization Algorithms

Metaheuristic algorithms are problem-independent optimization techniques designed to find the optimal solutions for real-world problems. Meta-heuristics can be separated into two main groups, i.e. population-based and single solution-based approaches. The population-based algorithms can be further classified into four groups[32], such as swarm intelligence-based, physics-inspired, human-based, and evolutionary-based algorithms, as presented in Fig. 8.

The techniques of swarm intelligence algorithms are derived from group actions seen in the natural world. The Physics-based algorithms make use of

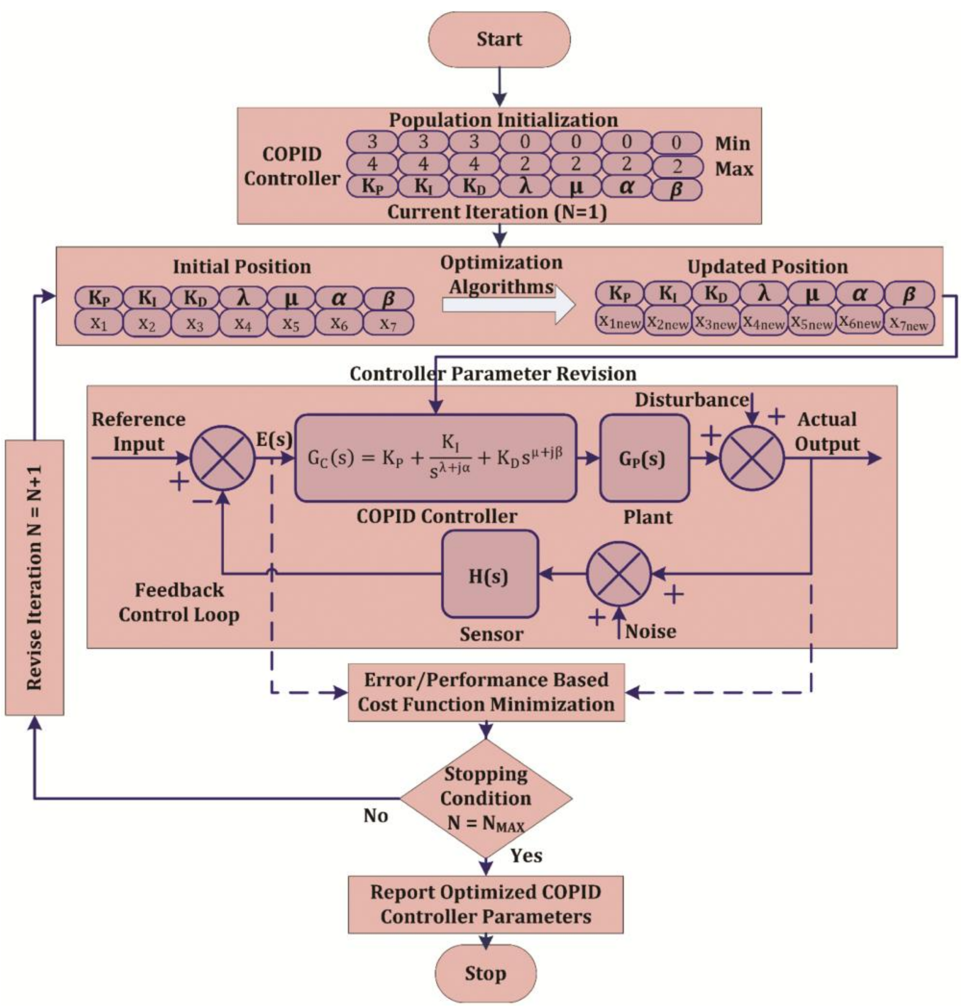

Fig. 7 — Parameter Estimation of COPID Controller using optimization techniques

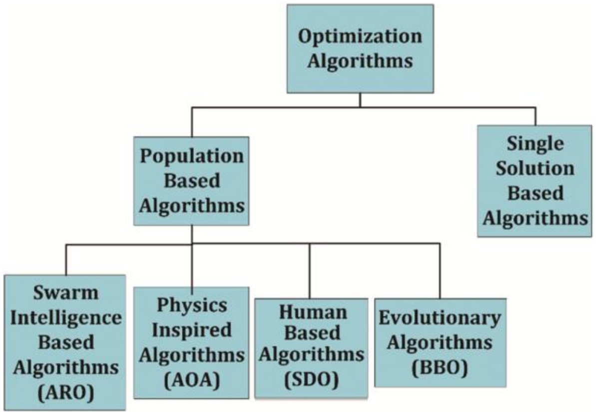

Fig. 8 — Optimization algorithm categorization

physics theories. The Evolutionary algorithms are created by taking inspiration from natural evolutionary processes. Likewise, human actions and choices have an impact on Human-based algorithms. Even though each of the algorithms has a unique searching method, population-based techniques share a dual-phase procedure of searching that includes both exploration and exploitation phases. Different nature-inspired population-based optimization algorithms used in this article for the estimation of optimal controller parameters are discussed below.

3.1 Artificial Rabbit Optimization (ARO*)*

The ARO is a population-based metaheuristic that draws inspiration from the survival techniques of rabbits. It belongs to the category of swarm intelligence-based algorithms. A potential solution that travels across the search space is represented by each rabbit in this algorithm. Rabbits use detour foraging behavior to explore, which enables them to randomly survey large areas. Exploitation is modelled through hiding behavior, where rabbits focus on refining solutions near the best location, which decreases over iterations. ARO avoids local optima by keeping movement directions unpredictable. ARO has relatively fewer control settings, and its computational complexity is modest. The execution of this algorithm can be understood from the pseudo-code given below.

In the pseudo-code for ARO, N stands for the population size, representing the total number of rabbits (candidate solutions). X_i denotes the position of the i^{th} rabbit in the search space, while X_{i_new} represents its updated position after exploration or exploitation. X_{best} indicates the best solution found so far, and X_{rand} It is a randomly selected rabbit used to guide exploration. The energy factor A controls the balance between exploration and exploitation over iterations; t is the current iteration number, and Max Iterations is the maximum number of iterations; r is a random number used to decide whether a rabbit performs exploration or exploitation; R is a random direction vector for hiding behavior, and $f(X_i)$ is the fitness value of the i^{th} rabbit. These variables collectively govern how the algorithm searches for the optimal solution across the search space. The pseudo code of the ARO algorithm is presented below.

1 Initialization
Initialize a population of rabbits X_i ($i = 1$ to N) randomly within
search space [lower_bound, upper_bound]
X_i = (upper_bound - lower_bound) × rand () + lower_bound
Evaluate the fitness value $f(X_i)$ for all the rabbits and identify the best rabbit X_{best}

2 Main Loop (Iterations)
For iteration t = 1 to Max Iterations
Do
Compute an energy factor ‘A’:
A = 4 × (1 - t / Max Iterations)

3 Decision: Exploration or Exploitation?
For each rabbit i = 1 to N do
Generate random number r ∈ [0, 1]
If r < 0.5 then
Detour Foraging Behavior (Exploration)
Select a random rabbit X_{rand} (rand ≠ i) and update position
$X_{i\_new} = X_i + A \times rand() \times (X_{rand} - X_i)$
Else
Random Hiding Behavior (Exploitation)
Generate random direction vector R ∈ [-1, 1] and update position
$X_{i\_new} = X_{best} + A \times R \times |X_{best} - X_i|$
End If
Apply boundary constraints to X_{i_new} and evaluate fitness $f(X_{i\_new})$
If $f(X_{i\_new}) < f(X_i)$ then
$X_i = X_{i\_new}$
End If
If $f(X_i) < f(X_{best})$ then
$X_{best} = X_i$
End If
End For
End For
Return X_{best}

3.2 Arithmetic Optimization Algorithm (AOA)

The AOA is a population-based physics-inspired metaheuristic algorithm that follows basic arithmetic operations. For exploration, it uses multiplication and division; for local exploitation, it uses addition and subtraction. The algorithm begins with randomly initialized candidate solutions within defined bounds. Each candidate solution is updated by applying mathematical operations based on the current best solution. Boundary conditions are applied to ensure that the generated solutions remain within feasible limits. The pseudo-code of the execution steps of this algorithm is provided below.

In the AOA, several variables and parameters are used to control the optimization process. 'N' represents the population size, which is the number of candidate solutions. X_i denotes the i^{th} solution vector in the population, while X_i^{new} is the updated solution generated during the search process. X_{best}, indicates the best solution found so far based on the fitness function $f(X_i)$, which evaluates the quality of each solution. The variables lower_bound and upper_bound define the minimum and maximum limits of the search space, and rand() generates a random number between 0 and 1. The algorithm runs for a maximum number of iterations defined by MaxIterations, with t representing the current iteration.

---

1 Initialization
Initialize population of solutions
X_i (i = 1 to N) randomly within search space
[lower_bound, upper_bound]

$$X_i = (\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}() + \text{lower\_bound}$$

Evaluate fitness $f(X_i)$for all solutions and identify the best solution X_{best}

2 Main Loop (Iterations)
For iteration $t = 1$ to Max Iterations
Do
Compute Math Optimizer Accelerated (MOA):$\text{MOA} = \text{MinMOA} + t \times \frac{(\text{MaxMOA} - \text{MinMOA})}{\text{MaxIterations}}$
Compute Math Optimizer Probability (MOP):$\text{MOP} = 1 - \left(\frac{t}{\text{MaxIterations}}\right)^{1/\alpha}$

3 Exploration or Exploitation Phase
For each solution $i = 1$to N
Do
Generate random number $r_1 \in [0,1]$

If $r_1 > MOA$
then
Exploration Phase
Generate random number $r_2 \in [0,1]$
If $r_2 < 0.25$then

$$X_i^{new} = X_{best} \div (\text{MOP} + \varepsilon) \times ((\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}() + \text{lower\_bound})$$

Else If $0.25 \le r_2 < 0.5$then

$$X_i^{new} = X_{best} \times \text{MOP} \times ((\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}() + \text{lower\_bound})$$

Else If $0.5 \le r_2 < 0.75$then

$$X_i^{new} = X_{best} - \text{MOP} \times ((\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}() + \text{lower\_bound})$$

Else

$$X_i^{new} = X_{best} + \text{MOP} \times ((\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}() + \text{lower\_bound})$$

End If
Else
Exploitation Phase
Generate random number $r_3 \in [0,1]$
If $r_3 < 0.5$then

$$X_i^{new} = X_{best} - \text{MOP} \times (X_{best} - X_i)$$

Else

$$X_i^{new} = X_{best} + \text{MOP} \times (X_{best} - X_i)$$

End If
End If
Apply boundary constraints to X_i^{new} and Evaluate fitness $f(X_i^{new})$
If $f(X_i^{new}) < f(X_i)$ then

$$X_i = X_i^{new}$$

End If
If $f(X_i) < f(X_{best})$ then

$$X_{best} = X_i$$

End If
End For
End For
Return X_{best}

Two important control parameters are Math Optimizer Accelerated (MOA) and Math Optimizer Probability (MOP), which regulate the transition between exploration and exploitation phases. MinMOA and MaxMOA define the range of MOA values, while α is a parameter that controls the convergence rate. A small constant ε is used to avoid division by zero. Additionally, random numbers r_1,

r_2, and r_3 are generated to decide whether the algorithm performs exploration or exploitation and to select the arithmetic operations used for updating solutions. The pseudo code of the AOA algorithm is provided below.

3.3 Supply Demand Optimization (SDO)

The SDO algorithm is a human-based algorithm that focuses on balancing customer demand with available supply efficiently and cost-effectively. SDO operates in two modes: stability mode and instability mode.

1 Initialization
Initialize population of solutions X_i ($i = 1$ to N)
randomly within search space
[lower_bound, upper_bound]

$$X_i = (\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}() + \text{lower\_bound}$$

Evaluate fitness $f(X_i)$ for all solutions
Identify the best solution X_{best}
Identify the worst solution X_{worst}

2 Main Loop (Iterations)
For iteration $t = 1$ to Max Iterations
Do
Compute demand weight:

$$D = 2\left(1 - \frac{t}{\text{Max Iterations}}\right)$$

Compute supply weight:

$$S = 2 \times \frac{t}{\text{Max Iterations}}$$

3 Supply–Demand Mechanism
For each solution $i = 1$ to N
Do
Generate random number $r \in [0,1]$
If $r < 0.5$ then
Demand Phase (Exploration)
Generate random solution X_{rand}

$$X_i^{new} = X_i + D \times \text{rand}() \times (X_{rand} - X_i)$$

Else
Supply Phase (Exploitation)

$$X_i^{new} = X_i + S \times \text{rand}() \times (X_{best} - X_i)$$

End If
Apply boundary constraints to X_i^{new} and evaluate fitness $f(X_i^{new})$
If $f(X_i^{new}) < f(X_i)$ then

$$X_i = X_i^{new}$$

End If
If $f(X_i) < f(X_{best})$ then

$$X_{best} = X_i$$

End If
If $f(X_i) > f(X_{worst})$ then

$$X_{worst} = X_i$$

End If
End For
End For
Return X_{best}

The supply-demand mechanism states that both the price and quantity of commodities can be encouraged to exploit the area around the equilibrium point by the stability mode, and the extent of this exploitation process varies over time. In contrast, the instability mode encourages exploration by pushing the price and quantity away from the equilibrium point, allowing the algorithm to investigate new regions of the search space. The pseudo-code of the SDO algorithm is mentioned above.

For the SDO method, let's say there are ‘n’ markets with ‘d ’ distinct commodity types, each of which has a specific quantity and price. The ‘d’ commodity quantities of a market are examined as a potential candidate solution, and the ‘d’ commodity prices of a market provide a candidate solution as ‘d’ variables to the optimization problem. This potential candidate solution takes the place of the candidate solution if it is superior. The overall procedure of this algorithm can be observed from the pseudo-code depicted below. In this algorithm, N indicates the size of the population, which is the total number of candidate solutions. X_i denotes the i^{th} solution, while X_i^{new} is the updated solution generated during the execution. The solutions are initialized within the search space limits defined by lower_bound and upper_bound, and rand() generates a random number between 0 and 1. The quality of each solution is evaluated by the $f(X_i)$ fitness function. X_{best} represents the best solution with the optimal fitness value, while X_{worst} indicates the worst solution in the population. The algorithm runs for a maximum number of iterations defined by MaxIterations, with t representing the current iteration. Two adaptive parameters, demand weight D and supply weight S, control the search behavior, where $D = 2(1 - t/\text{MaxIterations})$ mainly supports exploration and $S = 2(t/\text{MaxIterations})$ enhances exploitation as iterations progress. A random number r determines whether the algorithm performs the demand phase (exploration) using a randomly generated solution. X_{rand} or the supply phase (exploitation) that moves the solution toward the best solution X_{best}.

3.4 Bio Geographic Based Optimization (BBO)

A population-based evolutionary technique known as BBO is motivated by the spatial distribution of biological species. It shows potential solutions as habitats, where the habitat suitability index (HSI) represents the quality of the solution. Through migration mechanisms, high-quality solutions share features with lower-quality solutions. In contrast to conventional evolutionary algorithms, BBO prioritizes information exchange above reproduction.

To preserve variety and prevent premature convergence, mutation is used. By striking a balance between exploration and exploitation of the search space, the algorithm iteratively refines results. For complicated, nonlinear, and multidimensional optimization problems, BBO works especially well. It has been effectively used in power system optimization, control systems, and engineering. The approach shows good resilience and convergence properties. Overall, a versatile and effective framework for resolving practical optimization issues is offered by biogeography-based optimization. The operational flow of the algorithm is explained in the pseudo-code given below.

The BBO algorithm uses several variables to represent solutions and control the optimization process. The variable X_i denotes the i^{th} habitat (candidate solution) in the population, where $i = 1, 2, \ldots, N$, with N is the population size. Each habitat consists of D decision variables, represented by $X_i(k)$ where $k = 1,2,\ldots,D$. The lower_bound (LB) and upper_bound (UB) define the allowable range of each decision variable within the search space. The fitness function $f(X_i)$ evaluates the quality of each habitat. During migration, λ_i represents the immigration rate (probability of accepting features from other habitats), while μ_i represents the emigration rate (probability of sharing features with other habitats). Random numbers r_1, r_2, and r_3 are generated in the range $[0, 1]$ to control stochastic decisions in migration and mutation. The parameter P_{mod} is the modification probability that determines whether a decision variable will be replaced during migration, and P_{mut} is the mutation probability used to introduce random changes and maintain diversity. Finally, X_{best} stores the best habitat, i.e. the optimal solution. The pseudo-code of the BBO algorithm is discussed below.

1 Initialization
Initialize population of habitats
X_i ($i = 1$ to N)randomly within search space
[lower_bound, upper_bound]

$$X_i = (\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}() + \text{lower\_bound}$$

Evaluate fitness $f(X_i)$for all habitats
Sort habitats based on fitness (best → worst)
Identify the best habitat X_{best}

2 Main Loop (Iterations)
For iteration $t = 1$to Max Iterations Do
Compute immigration rate $\lambda_i$and emigration rate $\mu_i$for each habitat (based on habitat rank or fitness)

3 Migration Operation
For each habitat $i = 1$to Ndo
Generate random number $r_1 \in [0,1]$
If $r_1 < \lambda_i$then
Select a source habitat jprobabilistically according to μ_j
For each decision variable $k = 1$to Dimension do
Generate random number $r_2 \in [0,1]$
If $r_2 < P_{mod}$ then

$$X_i^{new}(k) = X_j(k)$$

Else

$$X_i^{new}(k) = X_i(k)$$

End If
End For
Else

$$X_i^{new} = X_i$$

End If

4 Mutation Operation
Generate random number $r_3 \in [0,1]$
If $r_3 < P_{mut}$ then
Select random decision variable k

$$X_i^{new}(k) = \text{lower\_bound} + (\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}()$$

End If
Apply boundary constraints to X_i^{new} and evaluate fitness $f(X_i^{new})$
If $f(X_i^{new}) < f(X_i)$then

$$X_i = X_i^{new}$$

End If
If $f(X_i) < f(X_{best})$then

$$X_{best} = X_i$$

End If
End For
End For
Return X_{best}

4 Simulation Studies & Statistical Analysis

Metaheuristic optimization algorithms are inherently stochastic and typically do not require an accurate or highly detailed mathematical model of the system. Owing to this advantage, they are particularly suitable for solving complex real-world engineering problems. In this study, four different population-based optimization algorithms, i.e. ARO, AOA, SDO, and BBO, are employed for the parameter estimation of the mentioned NIO cruise control systems. The performance of these algorithms is systematically evaluated to assess their effectiveness in achieving the desired control objectives. The parametric setting for the above optimization algorithms during the controller tuning procedure is provided in Table 2, and the results thus obtained are presented in the next section.

One of the primary concerns associated with optimization-based techniques is their stochastic nature, which may lead to variations in the obtained solutions across different runs. Hence, a statistical analysis is essential to account for this variability and to ensure the reliability of the obtained results. In this study, each optimization algorithm is executed independently 10 times, with 50 iterations in each run for every considered cost function. The statistical metrics obtained from these runs, namely the mean, standard deviation, and best values, are utilized to evaluate the consistency and effectiveness of the algorithms in determining the most appropriate solution.

Table 2 — Parameter settings in the optimization procedure

Parameter Settings	Values
Simulation Duration	100 s
Sampling Duration	0.01s
Total number of Search Agents	15
Total number of Iterations	50
Total number of Repetitions	10

4.1 ACC with PID Controller

The step response of an ACC system with a PID controller, considering different optimization algorithms and objective functions, is presented in Table 3. Out of the 10 repetitions, the results with the least mean values are considered the best results and their corresponding standard deviation and minimum values are mentioned here. It can be observed from Table 3 that, while ARO produces a minimal mean for the ISE function, the AOA provides a minimal mean for all other objective functions. The same can also be verified from the convergence curves provided in Fig. 9. Further, the PID parameters in accordance with the best values of various cost functions and corresponding time domain specifications are presented in Table 4. The optimal value of the response in terms of least T_R and T_S was obtained for ZLG with the AOA optimization algorithm, which can also be noticed from the step response comparison curves provided in Fig. 10.

4.2 ACC with FOPID Controller

The results obtained from the ACC system with the FOPID controller adopting various optimization techniques and cost functions are shown in Table 5. While the ARO provides optimal results with ITSE, ITAE, and ZLG cost functions, the SDO produces better results corresponding to ISE, IAE, IT^2SE, and IT^2AE error functions. This can also be verified from the convergence curve shown in Fig. 11. The FOPID controller parameters with respect to these optimal results are mentioned in Table 6. Further, the step response comparison as presented in Fig. 12 suggests superior performance in terms of minimal T_R and T_S, which corresponds to the ARO-tuned ZLG objective function.

Table 3 — A statistical study demonstrating the ACC system with a PID controller

Cost Function		ISE	IAE	ITSE	ITAE	IT^2SE	IT^2AE	ZLG
	Mean	0.8689	1.5581	0.5437	10.9842	1.4767	285.6808	0.2455
ARO	Std Dev	0.0005	0.0086	0.0026	0.5625	0.0443	1.0774	0.0012
	Best	0.8685	1.5476	0.5399	10.3612	1.4192	284.0010	0.2444
	Mean	0.9090	1.5434	0.5380	10.3107	1.3932	283.7690	0.2444
AOA	Std Dev	2.3405	0	0	1.8724	0	5.9918	0.0004
	Best	0.9090	1.5434	0.5380	10.3107	1.3932	283.7690	0.2440
	Mean	0.9095	1.5617	0.5438	10.7267	1.4478	286.2417	0.2452
SDO	Std Dev	0.0004	0.0134	0.0030	0.2950	0.0261	1.1227	0.0004
	Best	0.9091	1.5451	0.5387	10.4871	1.4036	284.2469	0.2446
	Mean	0.9098	1.5501	0.5397	13.7647	1.6563	288.6426	0.2449
BBO	Std Dev	0.0018	0.0145	0.0029	1.8210	0.6733	5.8218	0.0018
	Best	0.9090	1.5434	0.5381	10.3107	1.3932	283.7690	0.2440

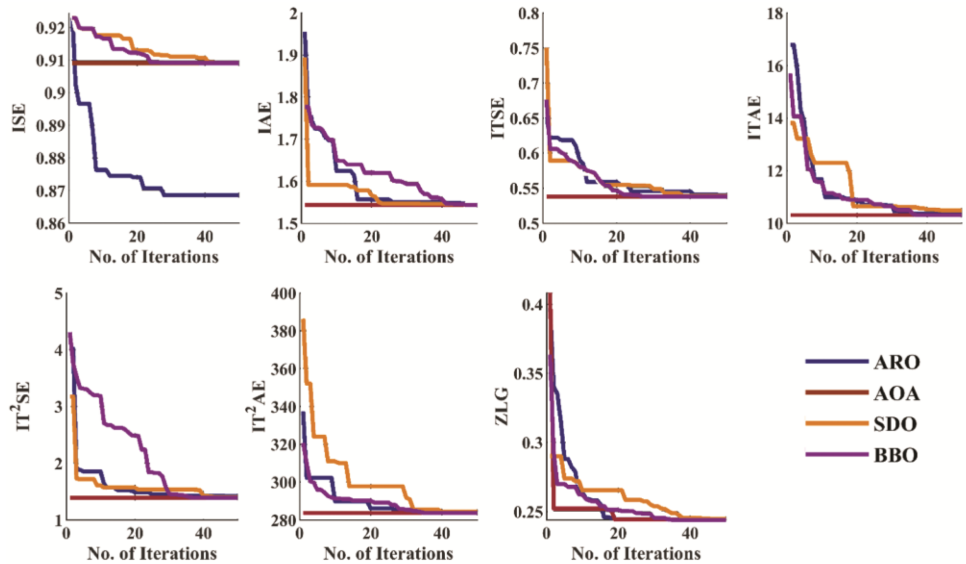

Fig. 9 — Convergence curves for different optimization algorithms with the PID controller

Table 4 — Time response analysis of the ACC system with PID controller

Cost Function	Controller Parameters (PID)	Tr	Ts	Mp (%ge)	Ess
ISE (ARO)	$K_P = 3.9991, K_I = 3.9991, K_D = 3.9991$	1.6682	2.7649	1.0171	-0.0031
IAE (AOA)	$K_P = 4, K_I = 0.1, K_D = 3$	1.6759	2.7979	0.7068	-0.0023
ITSE (AOA)	$K_P = 4, K_I = 0.1, K_D = 3$	1.6759	2.7979	0.7068	-0.0023
ITAE (AOA)	$K_P = 4, K_I = 0.1, K_D = 3$	1.6759	2.7979	0.7068	-0.0023
IT^2SE (AOA)	$K_P = 4, K_I = 0.1, K_D = 3$	1.6759	2.7979	0.7068	-0.0023
IT^2AE (AOA)	$K_P = 3, K_I = 0.3, K_D = 3$	2.1009	21.5179	9.3456	0.0006
ZLG (AOA)	$K_P = 4, K_I = 0.1352, K_D = 3$	1.6517	2.7052	1.6005	-0.0037

Table 5 — A statistical study demonstrating the ACC system with the FOPID controller

Cost Function		ISE	IAE	ITSE	ITAE	IT^2SE	IT^2AE	ZLG
	Mean	0.2775	0.5587	0.0854	0.4514	0.0397	10.2318	0.1035
ARO	Std Dev	0.0154	0.0115	0.0007	0.1175	0.0035	6.2231	0.0024
	Best	0.2571	0.5441	0.0838	0.2976	0.0367	2.6448	0.0999
	Mean	0.3056	0.7099	0.0962	3.9653	0.3464	108.1233	0.1193
AOA	Std Dev	0.0104	0.0340	0.0062	0.8317	0.1594	14.8485	0.0181
	Best	0.2925	0.6633	0.0888	2.8446	0.1649	81.2439	0.1018
	Mean	0.2722	0.5548	0.0858	0.4689	0.0387	4.7088	0.1038
SDO	Std Dev	0.0157	0.0090	0.0011	0.1382	0.0025	3.0238	0.0031
	Best	0.2543	0.5407	0.0842	0.3181	0.0349	1.4175	0.1008
	Mean	0.2891	0.6098	0.0879	1.6609	0.0664	33.4550	0.1072
BBO	Std Dev	0.0234	0.0240	0.0042	1.6560	0.0376	36.3847	0.0046
	Best	0.2598	0.5748	0.0846	0.2723	0.0361	0.7128	0.1000

4.3 ACC with TID Controller

Likewise, the results obtained from the ACC system with the TID controller are presented in Table 7. The AOA optimization algorithm has provided superior results for all the objective functions in this case. This can also be observed from Fig. 13 demonstrating all these objective functions and optimization techniques. Here, an identical performance is noticed for various cost functions such as ISE, IAE, ITSE, IT^2SE, and ZLG in terms of T_R, T_S, M_P, and E_{SS} as observed from Table 8. This can also be validated from the step response comparison curves presented in Fig. 14.

4.4 ACC with FOTID Controller

Similarly, the response of the ACC system with the FOTID controller considering different optimization algorithms and objective functions is obtained and demonstrated in Table 9. Here, while the ARO algorithm produced optimal results adopting ISE, IAE, IT^2SE, IT^2AE, and ZLG, the SDO algorithm

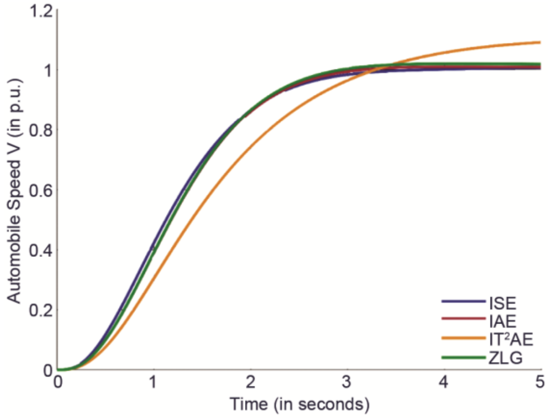

Fig. 10 — Step response of the ACC system with a PID Controller

provided better results for ITSE and ITAE, respectively. The convergence curve diagrams shown in Fig. 15 also validate the same. Further, the step response data corresponds to the optimal controller parameters mentioned in Table 10, which highlights the superiority of the ARO-tuned ZLG objective function in terms of minimal T_S and M_P. The step response comparison for different FOTID parameters, as shown in Fig. 16, also justifies the same.

4.5 ACC with COPID Controller

The results obtained by the application of the COPID controller for the ACC system, considering different optimization algorithms and performance indices, are provided in Table 11. It can be observed

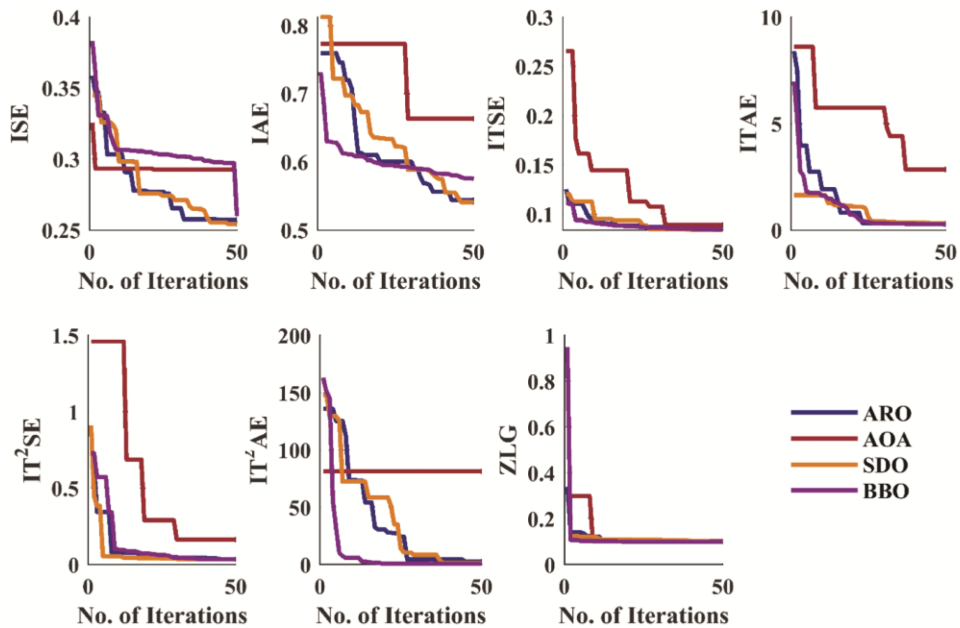

Fig. 11 — Convergence curves for different optimization algorithms with FOPID controller

Table 6 — Time response analysis of the ACC system with FOPID controller

Cost Function	Controller Parameters (FOPID)	Tr	Ts	Mp (%ge)	Ess
ISE (SDO)	$K_P = 3.9973, K_I = 0.1691, K_D = 3.9404$ $\lambda = 0.9969, \mu = 1.6076$	2.1201	9.9368	10.2065	0.0008
IAE (SDO)	$K_P = 3.9838, K_I = 0.1789, K_D = 3.9838$ $\lambda = 1.0040, \mu = 1.0087$	0.8096	1.2851	0.1283	0
ITSE (ARO)	$K_P = 3.9917, K_I = 0.1747, K_D = 3.9973$ $\lambda = 1.0062, \mu = 1.0525$	0.8651	1.6027	0.3054	0
ITAE (ARO)	$K_P = 3.6311, K_I = 0.1640, K_D = 3.7233$ $\lambda = 1.0004, \mu = 0.9953$	0.8772	1.4336	0	0
IT²SE (SDO)	$K_P = 3.9418, K_I = 0.1761, K_D = 3.9505$ $\lambda = 1.0025, \mu = 0.9953$	0.8039	1.2516	0.5842	0
IT²AE (SDO)	$K_P = 3.8282, K_I = 0.1749, K_D = 3.0350$ $\lambda = 1, \mu = 0.9762$	0.8898	2.8463	5.0232	0
ZLG (ARO)	$K_P = 3.8298, K_I = 0.1078, K_D = 3.9941$ $\lambda = 1.1457, \mu = 0.9671$	0.7961	1.2349	0.3744	-0.0025

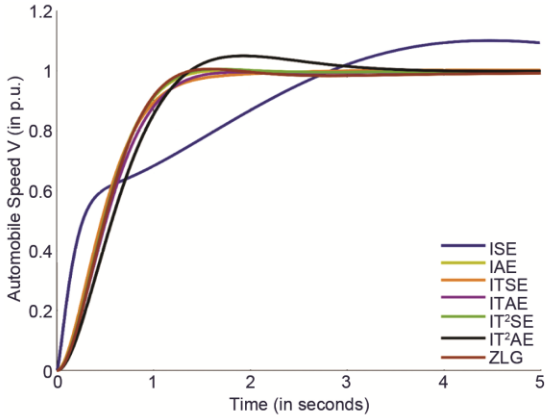

Fig. 12 — Step response of the ACC system with a FOPID Controller

from this comparison table that optimal results were obtained for the ARO-tuned ISE, ITSE, and IT^{2}AE, SDO-tuned IAE, ITAE, and IT^{2}SE, and BBO-tuned ZLG objective functions, respectively. The convergence curves for different algorithms presented in Fig. 17 also validate the same. The optimal values of the COPID controller corresponding to the various cost functions are presented in Table 12. The superiority of the BBO-tuned ZLG in terms of minimal T_R and T_S can be noticed from the same. This can again be validated from the step response comparison presented in Fig. 18.

As the optimal values of different IO and NIO controllers were obtained following the above

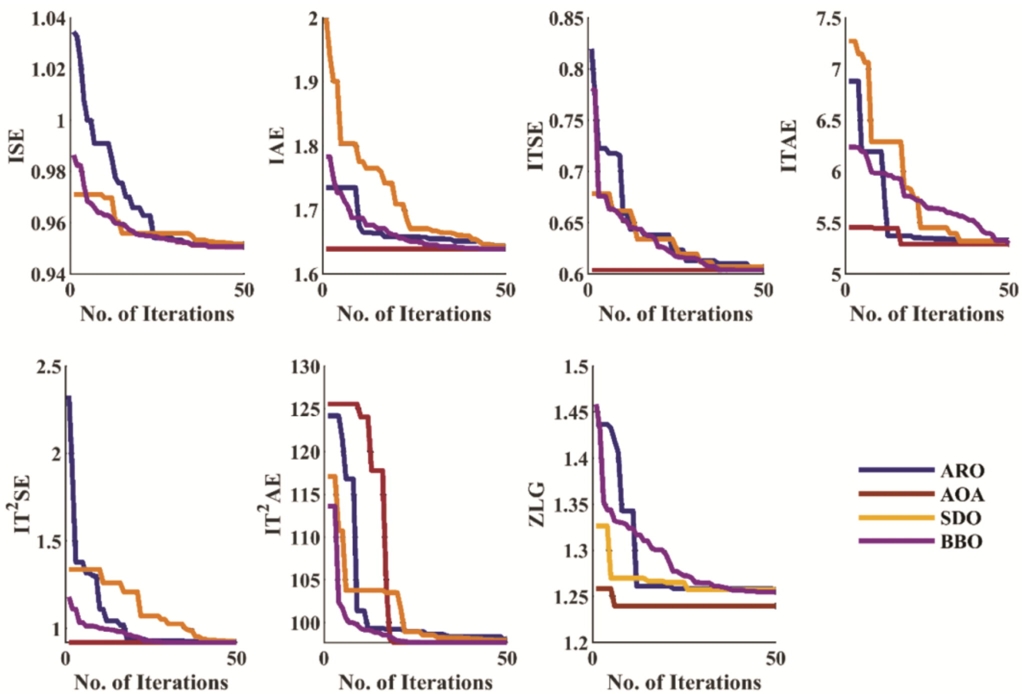

Fig. 13 — Convergence curves for different optimization algorithms with TID controller

Table 7 — A statistical study demonstrating the ACC system with the TID controller

Cost Function		ISE	IAE	ITSE	ITAE	IT^{2}SE	IT^{2}AE	ZLG
	Mean	0.9539	1.6538	0.6125	5.4850	0.9475	99.0506	1.2678
ARO	Std Dev	0.0014	0.0058	0.0044	0.2752	0.0186	0.5731	0.0085
	Best	0.9515	1.6466	0.6061	5.3301	0.9276	98.0544	1.2568
	Mean	0.9505	1.6394	0.6038	5.3033	0.9204	98.5150	1.2397
AOA	Std Dev	0	2.3405	1.1702	0.0163	2.3405	1.4752	2.3405
	Best	0.9505	1.6394	0.6038	5.2888	0.9204	97.6795	1.2397
	Mean	0.9543	1.6608	0.6143	5.4011	0.9669	98.6848	1.2649
SDO	Std Dev	0.0016	0.0133	0.0050	0.0624	0.0265	0.5817	0.0051
	Best	0.9517	1.6441	0.6069	5.3158	0.9275	97.9479	1.2572
	Mean	0.9573	1.6621	0.6188	6.2445	1.0356	101.6975	1.2824
BBO	Std Dev	0.0130	0.0319	0.0186	0.4458	0.1161	4.2110	0.0326
	Best	0.9505	1.6394	0.6038	5.296	0.9205	97.6781	1.2550

Table 8 — Time response analysis of the ACC system with TID controller

Cost Function	Controller Parameters (TID)	Tr	Ts	Mp (%ge)	Ess
ISE (AOA)	$K_P = 4, K_I = 0.1, K_D = 3, n = 10$	1.4477	6.6754	9.9776	0.0001
IAE (AOA)	$K_P = 4, K_I = 0.1, K_D = 3, n = 10$	1.4477	6.6754	9.9776	0.0001
ITSE (AOA)	$K_P = 4, K_I = 0.1, K_D = 3, n = 10$	1.4477	6.6754	9.9776	0.0001
ITAE (AOA)	$K_P = 4, K_I = 0.1059, K_D = 3, n = 10$	1.4458	6.7714	10.1123	-0.0001
IT^2SE (AOA)	$K_P = 4, K_I = 0.1, K_D = 3, n = 10$	1.4477	6.6754	9.9776	0.0001
IT^2AE (AOA)	$K_P = 4, K_I = 0.3, K_D = 3, n = 3.2798$	1.2113	11.4496	39.4099	-0.0002
ZLG (AOA)	$K_P = 4, K_I = 0.1, K_D = 3, n = 10$	1.4477	6.6754	9.9776	0.0001

Table 9 — A statistical study demonstrating the ACC system with a FOTID controller

Cost Function		ISE	IAE	ITSE	ITAE	IT^2SE	IT^2AE	ZLG
ARO	Mean	0.3226	0.6705	0.1011	1.1594	0.0673	13.4074	0.1195
	Std Dev	0.0209	0.0219	0.0018	0.0018	0.0025	2.4113	0.0045
	Best	0.2919	0.6419	0.0978	0.0978	0.0627	9.3040	0.1131
AOA	Mean	0.3297	0.7094	0.1096	2.6756	2.6756	35.7262	0.1294
	Std Dev	0.0137	0.0558	0.0090	0.7069	0.7069	12.2306	0.0106
	Best	0.2915	0.6416	0.0982	1.6538	1.6538	19.7280	0.1163
SDO	Mean	0.3285	0.6779	0.1000	1.0293	0.0679	13.9136	0.1205
	Std Dev	0.0147	0.0207	0.0017	0.1071	0.0081	1.3857	0.0059
	Best	0.2927	0.6374	0.0977	0.8480	0.0591	11.5685	0.1115
BBO	Mean	0.3423	0.7052	0.1070	1.5149	0.0900	17.2836	0.2059
	Std Dev	0.0078	0.0325	0.0057	0.7664	0.0445	11.8298	0.1721
	Best	0.3341	0.6585	0.0990	0.8979	0.0612	11.7189	0.1219

Table 10 — Time response analysis of the ACC system with FOTID controller

Cost Function	Controller Parameters (FOTID)	Tr	Ts	Mp (%ge)	Ess
ISE (ARO)	$K_T = 3.8739, K_I = 0.2687, K_D = 3.9232$ $\lambda = 0.5193, \mu = 1.6077, n = 9.9434$	1.4568	9.6219	12.2200	0.0098
IAE (ARO)	$K_T = 3.3260, K_I = 0.1027, K_D = 3.9964$ $\lambda = 1.1045, \mu = 0.9065, n = 9.9517$	0.8318	1.9038	2.1800	0
ITSE (SDO)	$K_T = 3.8287, K_I = 0.1018, K_D = 3.9961$ $\lambda = 1.1308, \mu = 0.9982, n = 9.7395$	0.8590	3.6831	2.3341	0
ITAE (SDO)	$K_T = 3.2143, K_I = 0.1503, K_D = 3.9524$ $\lambda = 1.0448, \mu = 0.7874, n = 9.8680$	0.7692	2.4803	9.2432	0
IT^2SE (ARO)	$K_T = 3.5866, K_I = 0.1585, K_D = 3.9577$ $\lambda = 1.0486, \mu = 0.8908, n = 9.9912$	0.7890	2.6190	5.5175	0.0001
IT^2AE (ARO)	$K_T = 3.2642, K_I = 0.1780, K_D = 3.5867$ $\lambda = 1.0310, \mu = 0.5746, n = 9.6254$	0.7319	4.3368	26.2570	0
ZLG (ARO)	$K_T = 3.0654, K_I = 0.1087, K_D = 3.9893$ $\lambda = 1.1686, \mu = 0.8973, n = 8.5198$	0.8698	1.3468	0.9414	-0.0015

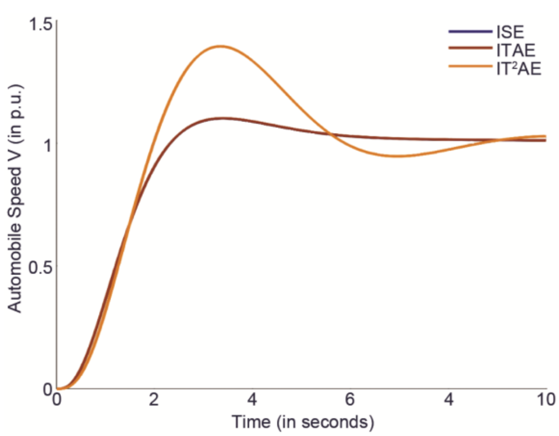

Fig. 14 — Step response of the ACC system with a TID Controller

optimization-based procedures, a comparison among all the controllers is also needed to identify the best possible controller for the performance improvement of the ACC system. A step response comparison among all the applied controllers is presented in Fig. 19, and the corresponding time specification-based statistical study is provided in Table 13. Both highlight the supremacy of the applied COPID controller over other controllers in terms of minimal T_R, T_S, and M_P. A frequency domain based bode plot comparison among all the applied controllers is also presented in Fig. 20. To justify the supremacy of the applied COPID controller as well as the procedure, some comparative studies with the existing literature

Table 11 — A statistical study demonstrating the ACC system with the COPID controller

Cost Function		ISE	IAE	ITSE	ITAE	IT^2SE	IT^2AE	ZLG
	Mean	0.3141	0.6120	0.0925	0.7624	0.0541	9.3202	0.1264
ARO	Std Dev	0.0106	0.0380	0.0021	0.3685	0.0245	3.9419	0.0210
	Best	0.2936	0.5647	0.0884	0.4130	0.0390	4.6510	0.1070
	Mean	0.3188	0.7890	0.1091	4.7081	0.4174	108.3112	0.5687
AOA	Std Dev	0.0022	0.0454	0.0139	0.6392	0.2370	24.5490	0.5111
	Best	0.3164	0.7108	0.0893	3.6855	0.1655	57.7830	0.1495
	Mean	0.3200	0.6018	0.0933	0.6352	0.0483	12.0336	0.2598
SDO	Std Dev	0.0017	0.0264	0.0044	0.2282	0.0044	5.2614	0.1820
	Best	0.3164	0.5573	0.0884	0.3527	0.0417	4.4427	0.1229
	Mean	0.3232	0.6283	0.0996	1.7285	0.0946	70.8154	0.1197
BBO	Std Dev	0.0080	0.0668	0.0158	2.4295	0.0404	60.2570	0.0113
	Best	0.3067	0.5504	0.0872	0.4399	0.0421	3.3910	0.1014

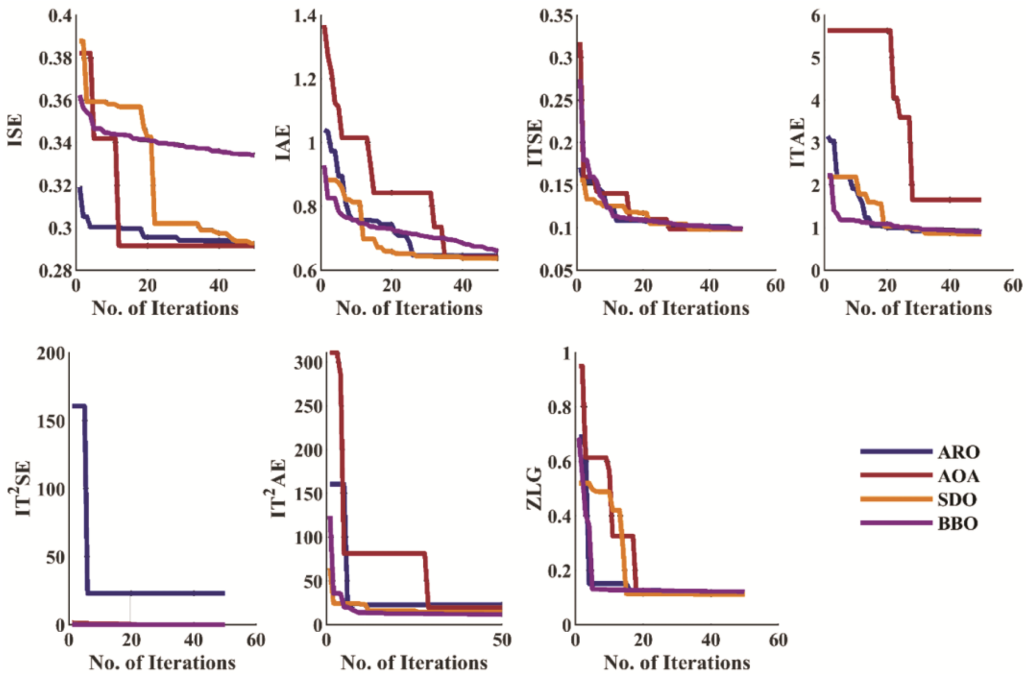

Fig. 15 — Convergence curves for different optimization algorithms with FOTID controller

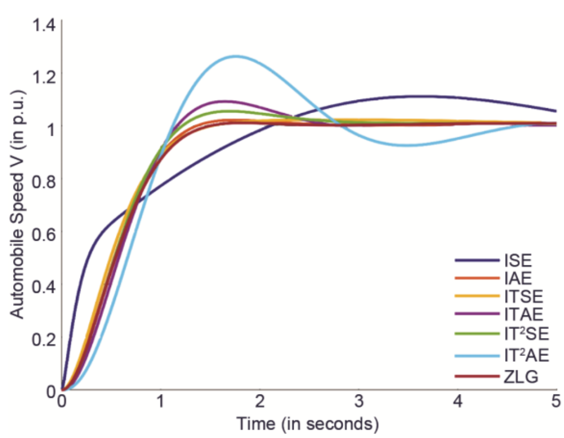

Fig. 16 — Step response of the ACC system with a FOTID Controller

are also provided in Table 13. The performance of the COPID controller is found to be better compared to other similar studies provided in the literature.

5 Robustness study

Ensuring the robustness of the designed controller against parameter variations, external disturbances, and measurement noise is a fundamental requirement for any closed-loop control system. In this study, the robustness of the COPID-based cruise controller for the ACC system is investigated under internal parameter variations, including changes in vehicle mass (M) and driver reaction time (τ); external disturbances represented by variations in road inclination (θ); and reference-tracking conditions

Table 12 — Time response analysis of the ACC system with COPID controller

Cost Function	Controller Parameters (COPID)	Tr	Ts	Mp (%ge)	Ess
ISE (ARO)	$K_P = 3.8584$, $K_I = 0.1838$, $K_D = 3.9739$ $\lambda = 1.2090$, $\mu = 1.7355$, $\alpha = 0.7169$, $\beta = 0.1379$	2.4763	3.2527	1.6888	-0.1028
IAE (SDO)	$K_P = 3.9571$, $K_I = 0.2148$, $K_D = 3.9398$ $\lambda = 1.0054$, $\mu = 1.0383$, $\alpha = 0.3969$, $\beta = 0.1026$	0.8650	1.4899	0.0482	-0.0019
ITSE (ARO)	$K_P = 3.9328$, $K_I = 0.1757$, $K_D = 3.9962$ $\lambda = 1.0286$, $\mu = 1.0318$, $\alpha = 0.3139$, $\beta = 0.1600$	0.8584	1.4474	0.0113	0.0001
ITAE (SDO)	$K_P = 3.2928$, $K_I = 0.1507$, $K_D = 3.9923$ $\lambda = 1.0010$, $\mu = 1.0093$, $\alpha = 0.1201$, $\beta = 0.3878$	1.0247	1.7801	0	0.0001
IT^2SE (SDO)	$K_P = 3.6818$, $K_I = 0.1797$, $K_D = 3.9978$ $\lambda = 1.0006$, $\mu = 0.9746$, $\alpha = 0.2496$, $\beta = 0.2406$	0.8394	1.2923	0.9983	0.0025
IT^2AE (ARO)	$K_P = 3.1289$, $K_I = 0.1910$, $K_D = 3.5618$ $\lambda = 1.0018$, $\mu = 1.1187$, $\alpha = 0.4637$, $\beta = 0.5455$	1.4125	3.5073	2.0944	-0.0164
ZLG (BBO)	$K_P = 3.5728$, $K_I = 0.1685$, $K_D = 4$ $\lambda = 1.1539$, $\mu = 0.9344$, $\alpha = 0.4550$, $\beta = 0.1006$	0.7884	1.1937	1.5963	0.0064

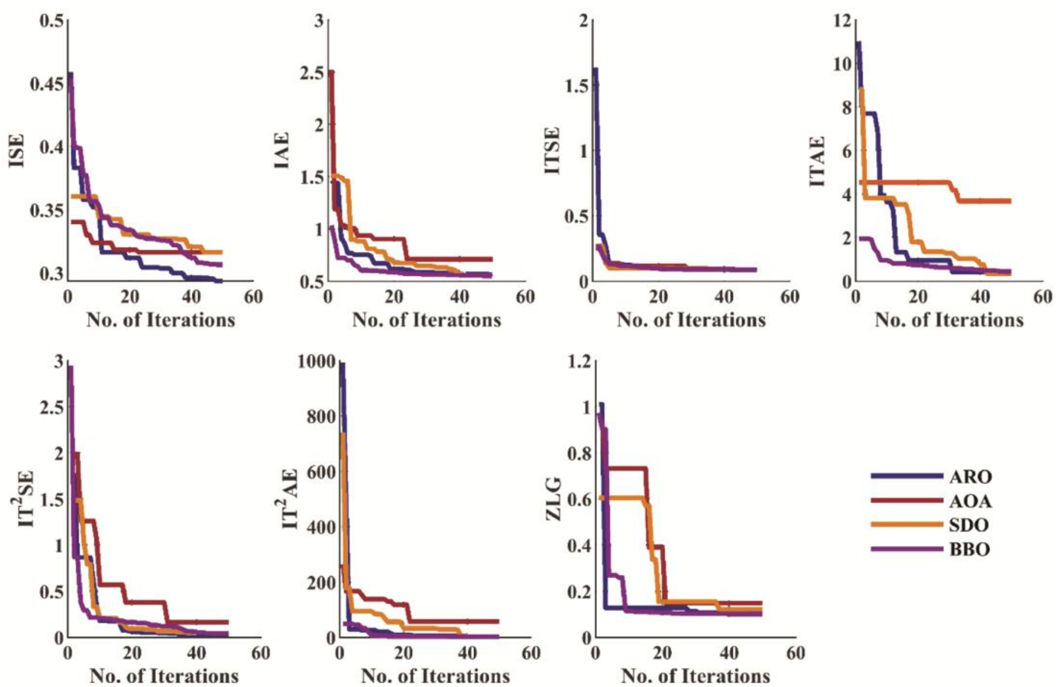

Fig. 17 — Convergence curves for different optimization algorithms with COPID controller

involving variations in the vehicle set speed ($\mathcal{V}$), as discussed below. The robustness analysis considers parameter variations of ±10 %, ±20 %, and ±30 %, representing small, moderate, and relatively severe deviations from nominal operating conditions, respectively, which may be encountered in practical cruise control systems.

5.1 Variations in Automobile Mass (M)

This analysis involves the variations in automobile mass over a range $1050\,\text{kg} \le \text{M} \le 1950\,\text{kg}$, i.e. a $\pm(10\,\% - 30\,\%)$ variation from the reference mass of 1500 kg. The corresponding response on the system output is presented in Fig. 21, which validates the efficient control of COPID towards addressing the effects of these parameter variations. The performance of COPID under both the light load, i.e. -30 %, as well as the heavy load, i.e. $+30$ %, is satisfactory.

5.2 Variations in Driver Reaction Time (τ)

Again, the response of the COPID-powered ACC system towards the variations in driver reaction time (τ) is evaluated and presented in Fig. 22. Similar $\pm(10\,\% - 30\,\%)$parameter variations, i.e. over the range $0.14\ \mathrm{s} \leq \tau \leq 0.26\ \mathrm{s}$, are considered here to check the effectiveness of the applied COPID controller. It can be noticed from Fig. 22 that the COPID-based ACC performs efficiently under faster as well as slower reaction times.

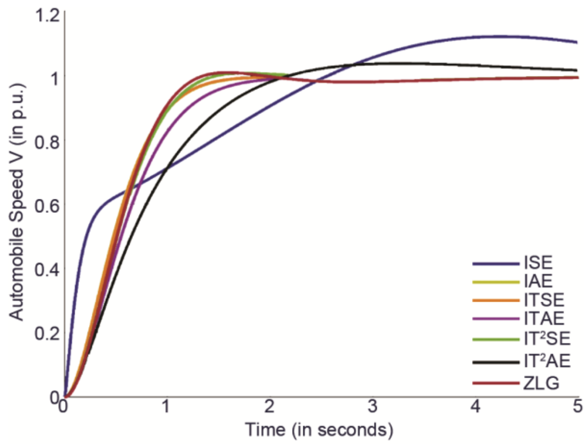

Fig. 18 — Step response of the ACC system with a COPID Controller

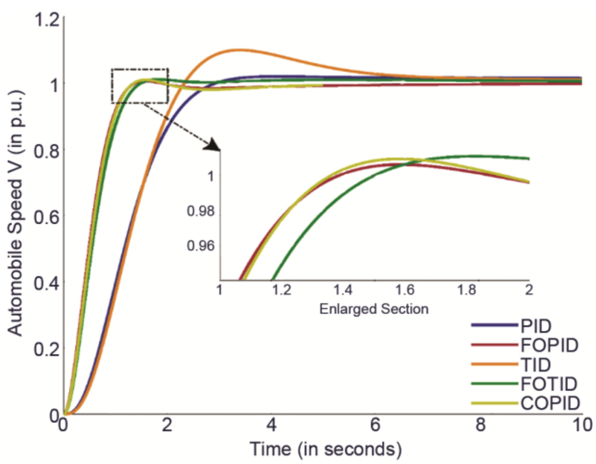

Fig. 19 — Step response comparison among different controllers with the ACC system

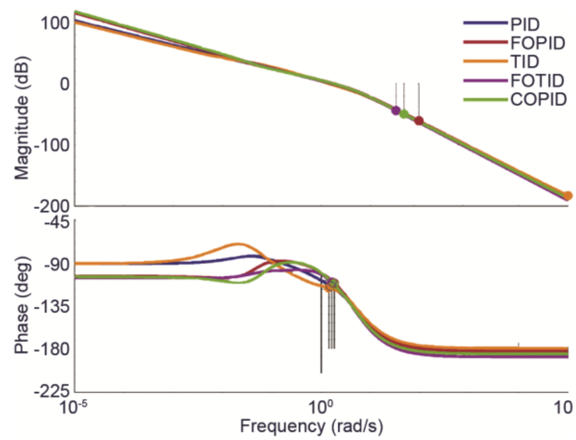

Fig. 20 — Bode plot comparison among different controllers with the ACC system

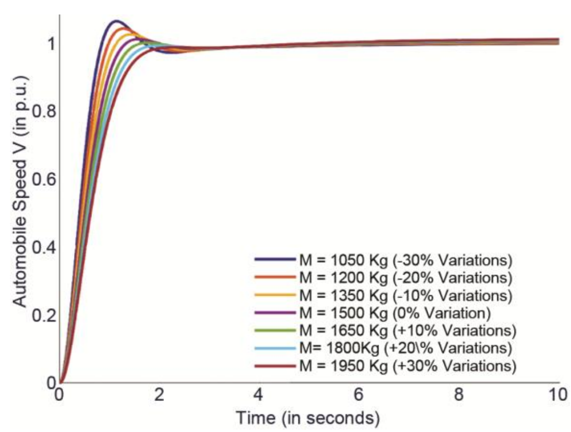

Fig. 21 — Output of the COPID-based ACC system under variations in automobile mass

Table 13 — Time response analysis of the ACC system with the best values of all the controllers

Cost Function	Time Domain Specifications			
	Tr	Ts	Mp (%ge)	Ess
PID (ZLG-AOA)	1.6517	2.7052	1.6005	-0.0037
FOPID (ZLG-ARO)	0.7961	1.2349	0.3744	-0.0025
TID (ZLG-AOA)	1.4477	6.6754	9.9776	0.0001
FOTID (ZLG-ARO)	0.8698	1.3468	0.9414	-0.0015
COPID (ZLG-BBO)	0.7884	1.1937	1.5963	0.0064
PID [2]	0.8037	1.2574	0.4146	0
PID [3]	0.8094	1.7320	0.6120	0
PID [4]	0.8161	1.2580	0.1093	0
PID [14]	0.8500	1.3160	0.9200	0
FOPID [19]	0.8066	1.8292	2.0597	0

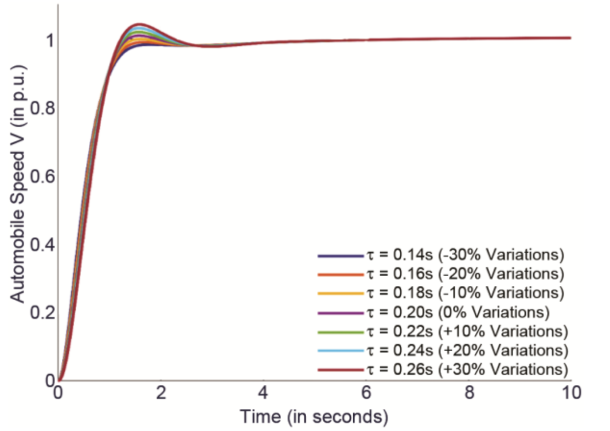

Fig. 22 — Output of the COPID-based ACC system under variations in driver reaction time

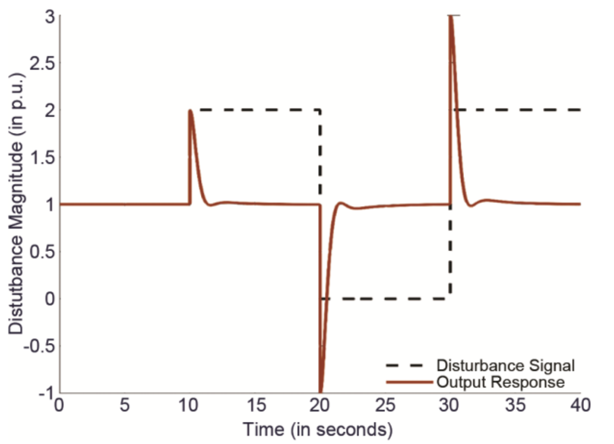

Fig. 23 — Output of the COPID-based ACC system under variations in road inclination

5.3 Variations in Road Inclination (θ)

This analysis includes the changes in the road inclinations as an external disturbance applied over the COPID-based ACC system. The road inclination angle (θ) is included in the nonlinear ACC system dynamics, i.e. in Eq. (1) and neglected in the linearized ACC transfer function model, i.e. in Eq. (4). Therefore, the effect of a change in road inclination can only be observed considering the nonlinear ACC system model. The allowable variations in road inclination are within a range $(4^{\mathrm{o}} - 6^{\mathrm{o}})$.

It is difficult to analyze the effect of road inclination from the present linearized ACC system with the COPID controller. This study models these variations as an external applied disturbance to the ACC system with a variation of $(2\,\mathrm{p.u} - 4\,\mathrm{p.u})$ and verifies the effect of the COPID controller towards addressing the same. Satisfactory disturbance rejection property of the COPID controller for the linearized ACC system can be noticed from Fig. 23.

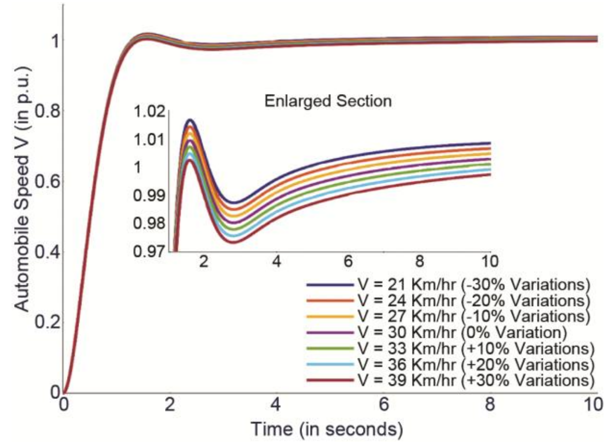

Fig. 24 — Output of the COPID-based ACC system under variations in reference speed

5.4 Variations in reference speed ($\mathcal{V}$)

The input reference tracking characteristics of the COPID cruise controller are also tested in this article by providing a variation of $\pm(10\,\% - 30\,\%)$ in the set-point automobile speed, and the corresponding response is presented in Fig. 24. Acceptable cruise control characteristics are also noticed for the COPID controller under these applied variations.

6 Conclusion

This study investigates the performance of various integer and NIO controllers for the ACC system under varying operating conditions. The primary focus of this work is the analysis, design, and implementation of the COPID controller. Simulation results, supported by statistical analysis, demonstrate that the COPID controller outperforms the other considered controllers in terms of reference tracking accuracy, robustness, and disturbance rejection capability. Among the optimization algorithms employed, no single technique consistently dominates across all evaluation metrics. However, in terms of the selected performance indices, the ZLG objective function exhibits superior performance by achieving reduced settling time, minimal overshoot, and lower tracking error.

Furthermore, the ZLG-based BBO-tuned COPID controller shows improved performance not only compared with the integer and NIO controllers considered in this work, such as PID, TID, FOPID, and FOTID, but also with several related approaches reported in the literature. Overall, the obtained results highlight the strong potential of COPID-based ACC design for enhancing vehicle speed regulation and ride comfort. Future work will focus on experimental validation and real-time implementation of the proposed control strategy to further establish its practical applicability.

Although the proposed methodology has been demonstrated for an automobile cruise control system, the underlying optimization framework is generic and can be readily extended to other engineering control applications. Future work will investigate frequency-domain optimization of COPID controllers by incorporating loop-shaping constraints, gain crossover frequency, phase margin, and iso-damping properties into the optimization framework, thereby combining the advantages of time-domain and frequency-domain controller design.

References

1 Heybetli F, Danayiyen Y, Taşdemir A B & Şenyiğit Ş, *Electric*, 25 (1) (2025) 1.
2 İzci D, Ekinci S, Kayri M & Eker E, *Electric*, 21 (3) (2021) 283.
3 Pradhan R, Majhi S K, Pradhan J K & Pati B B, *Eng J*, 21 (5) (2017) 347.
4 Pradhan R, Majhi S K, Pradhan J K & Pati B B, *J Indust Inform Integr*, 9, (2018) 45.
5 Izci D, Ekinci S, Kayri M & Eker E, *Evolving Syst*, 13 (3) (2022) 453.
6 Ekinci S & Izci D, *Evolution Intell*, 16 (4) (2023) 1339.
7 Ekinci S, Izci D, Abualigah L & Zitar R A, *J Bionic Eng*, 20 (4) (2023) 1828.
8 Ekinci S, Izci D, Abualigah L, Hussien A G, Thanh C L & Khatir S, *Int J Comput Intell Syst*, 16 (1) (2023) 129.
9 Chirantan S, Prusty C & Pati B B, *Eng Res Express*, 8 (1) (2026) 015303.
10 Çakan A, *Int J Automot Engineer Technol*, 15 (1) (2026) 20.
11 Zainuddin M A S, Abdullah M, Ahmad S & Tofrowaih K A, *Int J Automot Mechanic Eng*, 19 (1) (2022) 9460.
12 Izci D & Ekinci S, in *Proceedings of the 2nd International Conference on Artificial Intelligence: Advances and Applications (ICAIAA 2021)* (Springer Nature, Singapore) 2022 pp. 333–343.
13 Hlangnamthip S, Thammarat C, Sinsukudomchai C & Puangdownreong D, in *Proceedings of the 2024 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Comput Telecommun Eng*, (IEEE), 2024 pp. 108–111.
14 Turan A, *J Radiat Res Appl Sci*, 17 (1) (2024) 100810.
15 Polubny I, *IEEE Trans Automat Control*, 44 (1999) 208.
16 Monje C A, Chen Y, Vinagre B M, Xue D & Feliu V, *Fractional-order systems and controls: fundamentals and applications* (Springer, London), (2010).
17 Kerketta S, Pati T K, Dash J, Mishra S & Dakua B K, in *Proceedings of the 5th IEEE International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC)* (IEEE) 2025, pp. 1–6.
18 Abualigah L, Ekinci S, Izci D & Zitar R, *Intell Automat Soft Comput*, 38 (2) (2023) 169.
19 Idir A, Zemmit A, Khettab K, Nesri M, Guedida S & Canale L, *Sci Eng Technol*, 5 (1) (2025) 1.
20 Lurie, B. J, US-PATENT-APPL-SN-023253, Three-parameter tunable tilt-integral-derivative (TID) controller, 1994.
21 Dakua B K & Pati B B, *Eng Res Express*, 6 (3) (2024) 035331.
22 Dakua B K & Pati B B, *Math Model Control*, 4 (2024) 374.
23 Abdulwahhab O W, 99 (2020) 154.
24 Bin Roslan M N, Bingi K, Devan P A M & Ibrahim R, *Appl Syst Innov*, 7 (3) (2024) 33.
25 George M A, Elwakil A S, Allagui A & Psychalinos C, *IEEE Access*, 11 (2023) 118606.
26 Shahiri M, Ranjbar A, Karami M R & Ghaderi R, *Nonlinear Dynamics*, 84 (3) (2016) 1813.
27 Saleem O & Iqbal J, *Measure Control*, 56 (9) (2023) 1811.
28 Shah P, Sekhar R, Iswanto I & Shah M, *Adv Sci Technol Eng Syst J*, 6 (2021) 541.
29 Warrier P & Shah P, *J Robot Control*, 4 (3) (2023) 243.
30 Wolpert D H & Macready W G, *IEEE Trans Evolution Comput*, 1 (1) (2002) 67.
31 Gaing Z L, *IEEE Trans Energy Convers*, 19 (2) (2004) 384.
32 Baihan A, Ghith E, Garg H, Mirjalili S, Izci D, Rashdan M, Salman M & Saleem K A, *Int J Comput Intell Syst*, 18 (1) (2025) 86.
33 Wang L, Cao Q, Zhang Z, Mirjalili S & Zhao W, *Eng Appl Artif Intell*, 114 (2022) 105082.
34 Zhao W, Wang L & Zhang Z, *IEEE Access*, 7 (2019) 73182.
35 Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M & Gandomi A H, *Computer Methods Appl Mechanic Eng*, 376, 113609 (2021).
36 Simon D, *IEEE Trans Evolutionary comput*, 12 (6) (2008) 702.
37 Petráš I, *Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation* (Springer Science & Business Media, Berlin) 2011, p. 218.
38 Tepljakov, A, Petlenkov E & Belikov J, *Int J Microelectron Comput Sci*, 2 (2) (2011).