

Adaptive Metaheuristic Tuning of Fractional-Order PID Controllers for Enhanced Vehicle Cruise Control

Salahiddin H. Altahat¹, Iqbal M. Batiha^{2,3,*}, Obaida M. Aljabari⁴

¹*Department of Data Science and Artificial Intelligence, Al-Zaytoonah University of Jordan ,11733 Amman, Jordan*

²*Department of Mathematics, Al Zaytoonah University of Jordan, Amman 11733, Jordan*

³*Nonlinear Dynamics Research Center (NDRC), Ajman University, Ajman, UAE*

⁴*Department of Mathematics, University of Jordan, 11942 Amman, Jordan*

Abstract In recent years, the automotive industry has increasingly focused on developing advanced cruise control systems (CCS) to enhance vehicle safety, comfort, and efficiency. However, the dynamic nature of vehicle motion affected by factors such as road slope, aerodynamic drag, and load variations makes it difficult for conventional controllers to maintain stable performance under varying conditions. This study investigates the implementation of a Fractional-Order PID (FOPID) controller, which generalizes the classical PID by introducing fractional differentiation and integration orders. These additional parameters increase the controller's flexibility and allow for more accurate tuning of system dynamics, however, they increase system complexity. The fractional operators are implemented using Oustaloup Recursive Approximation and Continued Fraction Expansion (CFE). The novelty of this paper is to perform a comparative analysis that is conducted between classical PID and FOPID controllers, where the parameters of each are optimized using the Genetic Algorithm (GA) in vehicle CCS. Simulation results, based on a linearized Vehicle Cruise Control System (VCCS) model, show that the PID controller achieves better performance in step response and disturbance rejection, however, the FOPID Oustaloup presents a good step response and disturbance rejection. The CFE method has high steady state error and oscillation in most of its explored solutions by the GA algorithm.

Keywords Vehicle cruise control system (VCCS), Fractionalized order PID controller(FOPID), Genetic Algorithm(GA), Continued Fraction Expansion (CFE), Oustaloup Approximation (OA)

AMS 2010 subject classifications: 05C72, 05C15

DOI: 10.19139/soic-2310-5070-3441

1. Introduction

Automotive engineering has increasingly focused on developing advanced cruise control systems that demand exact and adaptable controllers to meet modern technological requirements [1, 2]. The dynamic behavior of vehicles is inherently complex and continuously varying, introducing significant nonlinearities into the control system. Such nonlinear characteristics often limit the effectiveness of conventional Proportional–Integral–Derivative (PID) controllers under diverse operating conditions[3, 4, 5].

To address these challenges, this study investigates the application of a Fractional-Order Proportional–Integral–Derivative (FOPID) controller [6, 7, 8]. This advanced control strategy extends the classical PID framework by incorporating fractional-order differentiation and integration operators. These additional tuning parameters enhance flexibility and precision, enabling superior performance for nonlinear and complex systems.

*Correspondence to: Iqbal M. Batiha (Email: i.batiha@zuj.edu.jo). Department of Mathematics, Al Zaytoonah University of Jordan, Amman 11733, Jordan.

For the practical realization of fractional operators in the frequency domain, two established approximation techniques—Continued Fraction Expansion (CFE) and the Oustaloup Approximation—are employed.

A detailed comparative analysis is conducted between classical PID and fractional-order PID controllers, with their parameters optimized using Genetic Algorithms (GA) and other metaheuristic optimization methods. Simulation results on a linearized Vehicle Cruise Control System (VCCS) model demonstrate that the FOPID controllers offer significant improvements in transient response, particularly in reducing rise and settling times[9, 10]. Among the optimized designs, the GA-based FOPID controller exhibits the best overall performance, achieving an optimal trade-off between rapid response and steady-state stability while maintaining robustness against parameter variations and external disturbances[11, 12, 13].

2. Fractional-order PID-Controller

The FoPID controller is used in various fields such as industry, medicine, and economics to improve and develop system performance. The concept of the fractional-order PID (FOPID) controller was introduced by Podlubny et al. in 1997 [14, 15, 16]. This controller extends the conventional PID by introducing two additional parameters, γ and ρ , alongside the traditional gains K_p , K_i , and K_d . The FOPID controller is defined by the following fractional-order integro-differential equation:

$$y(t) = K_p e(t) + K_i I^\gamma e(t) + K_d D^\rho e(t), \quad (1)$$

where I^γ is the Riemann-Liouville fractional integral of order γ , D^ρ is the Caputo fractional derivative of order ρ , and $e(t)$ is the error signal. By utilizing the forward Laplace transform of 1, we obtain the transfer function of the FOPID controller as follows[17]:

$$C(s) = \frac{Y(s)}{E(s)} = K_p + \frac{K_i}{s^\gamma} + K_d s^\rho, \quad (2)$$

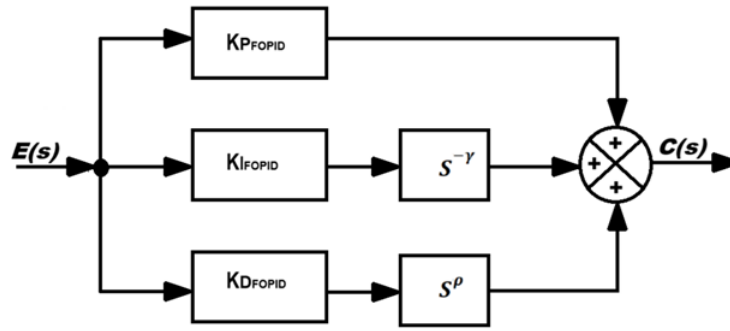
where $C(s)$ is the controller transfer function, $Y(s) = \mathcal{L}\{y(t)\}$, and $E(s) = \mathcal{L}\{e(t)\}$ is the Laplace transform of $y(t)$ and $e(t)$ respectively.

The primary objective of this research is to design and optimize a *Vehicle Cruise Control System (VCCS)* using a *Fractional-Order Proportional–Integral–Derivative (FOPID)* controller, in place of the conventional PID controller. The aim is to enhance control performance by minimizing rise time and settling time, while simultaneously improving system robustness and adaptability under nonlinear and time-varying operating conditions.

To achieve this goal, frequency-domain approximation techniques such as the *Oustaloup Approximation (OA)* and the *Continued Fraction Expansion (CFE)* are employed to realize the fractional-order differentiation and integration operators in a practical manner. Moreover, metaheuristic optimization algorithms, particularly the *Genetic Algorithm (GA)*, are utilized to effectively tune the controller parameters. This research demonstrates that the GA-optimized FOPID controller surpasses the traditional PID controller in terms of responsiveness, steady-state accuracy, and robustness against external disturbances and system parameter variations. The performance of the closed-loop system is evaluated using four standard performance metrics: steady-state error, rise time, settling time, and maximum overshoot. The *Genetic Algorithm (GA)* is then employed to explore the search space and obtain optimal values for the five FOPID controller parameters: K_p , K_i , K_d , γ , and ρ .

For comparison, the same optimization procedure is applied to the classical PID controller, where fractional approximation methods such as CFE and OA are unnecessary. In this case, the GA searches for optimal values of the three controller parameters: K_p , K_i , and K_d . Each optimization experiment is repeated ten times for both controllers to ensure the reliability of results and to provide the GA with sufficient opportunity to locate the global optimum.

The controller block diagram of the fractional order PID (FOPID) is illustrated in Figure 1. Both CFE and OA methods provide practical and accurate integer-order approximations of the fractional-order Laplace operators (s^γ and s^ρ), where $0 < \gamma, \rho < 1$ [12, 15, 18, 19].

Figure 1. Block diagram of the fractional-order $PI^\gamma D^\rho$ controller.

3. Literature Review-FOPID Controller

The transfer function $G(s)$ for the Vehicle Cruise Control System (VCCS) is given by equation 3 [20]:

$$G(s) = \frac{2.4767}{s^3 + 6.0476s^2 + 5.2856s + 0.238} \quad (3)$$

In this regard, we aim to approximate the fractional-order operators, represented in the Laplace domain as $s^{-\gamma}$ and s^ρ , using *Oustaloup Approximation (OA)* and the *Continued Fraction Expansion (CFE)*. This process results in two fractional-order PID controllers, $C_i(s)$, each associated with a closed-loop system $H_i(s)$, where $i = 1, 2$. These closed-loop systems are then evaluated and compared to identify the best-performing controller among the proposed designs.

We can express the closed-loop transfer function as

$$H(s) = \frac{C(s)G(s)}{1 + C(s)G(s)}, \quad (4)$$

where $H(s)$ is the closed-loop transfer function, $C(s)$ is the controller transfer function, and $G(s)$ is the plant transfer function.

3.1. Vehicle Cruise Control System (VCCS)

A Vehicle Cruise Control System (VCCS) is a feedback control system designed to maintain the vehicle speed at a desired reference value by automatically adjusting the throttle input. Although a VCCS consists of a speed sensor, a controller, an actuator, and the vehicle dynamics, its primary challenge arises from the nonlinear and time-varying behavior of the vehicle.

The performance of a cruise control system is influenced by several factors, including road slope variations, aerodynamic drag, rolling resistance, changes in vehicle load, and external disturbances. These factors introduce uncertainties and nonlinearities that may degrade the performance of conventional PID controllers, resulting in increased overshoot, longer settling times, and steady-state tracking errors.

To address these challenges, this study employs a Fractional-Order PID (FOPID) controller whose parameters are optimized using a Genetic Algorithm (GA). The additional tuning flexibility provided by fractional integration and differentiation orders enables improved transient response, more accurate speed tracking, and enhanced robustness against system uncertainties under varying operating conditions.

System Components

A typical VCCS consists of several key components that work in a closed-loop feedback control structure:

1. **Setpoint Input:** The driver interface (usually buttons on the steering wheel: ON, OFF, SET, RESUME, +, -) that allows the driver to engage the system and select the desired speed (the *reference* or *setpoint*).
2. **Speed Sensor:** A sensor (e.g., a wheel speed sensor or transmission sensor) that continuously measures the vehicle's actual speed. This is the *measured output* of the system.
3. **Controller:** The system's brain, typically an Electronic Control Unit (ECU). It compares the measured speed with the setpoint speed and calculates the necessary corrective action based on a control law (e.g., a **Proportional-Integral (PI)** or **PID** controller).
4. **Actuator:** The mechanism that executes the controller's command. In modern vehicles, this is an electronic throttle body that adjusts the throttle plate position, thereby controlling the engine's air intake and power output.
5. **Plant:** The vehicle itself, which is the dynamic system being controlled. The plant's output is the vehicle speed, which is influenced by the throttle input and external disturbances.

Control Loop Principle

The VCCS operates as a classic **closed-loop feedback control system**. The fundamental principle can be described by the following block diagram and equation:

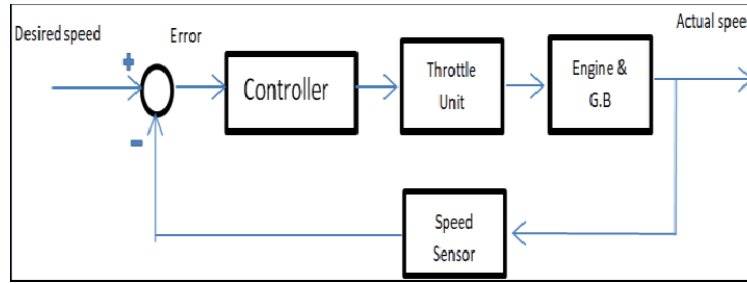


Figure 2. Block diagram of a typical Cruise Control System.

The controller's objective is to minimize the **error** $e(t)$, which is defined as:

$$e(t) = V_{\text{set}} - V_{\text{actual}}(t) \quad (5)$$

where V_{set} is the desired speed and $V_{\text{actual}}(t)$ is the measured speed at time t .

A standard **PID Controller** calculates the throttle command $u(t)$ as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (6)$$

where:

- K_p is the **Proportional gain**: Provides a reaction based on the current error.
- K_i is the **Integral gain**: Eliminates steady-state error (e.g., a persistent slight speed difference on an incline).
- K_d is the **Derivative gain**: Dampens the system's response and reduces overshoot by considering the rate of change of the error.

Key Advantages

- Reduces driver fatigue on long trips.
- Can help improve fuel economy by maintaining a constant speed.
- Helps drivers avoid unintentionally exceeding speed limits.
- (For ACC) Enhances safety by providing automated braking and distance keeping.

3.2. Genetic Algorithms

Genetic algorithms (Genetic Algorithms - GAs) are random search and optimization techniques inspired by the principles of natural evolution and biological heredity. These algorithms belong to the family of evolutionary algorithms, are known for their effectiveness in solving complex problems that traditional methods often fail to cope with.

The basic idea of genetic algorithms is to simulate the process of natural selection by developing a set of possible solutions over successive generations. Each solution usually represents a chromosome, and assesses its quality using the cost Function to determine its suitability for solving the problem. Evolutionary processes are then applied to the solutions to discover the best solutions and efficiently exploit the search space [21, 22, 23].

Fundamental Principles of Genetic Algorithms

1. **Representation (Encoding):** Solutions are represented in a structured form, most commonly as fixed-length binary strings. Each position (bit) in the string corresponds to a specific feature of the solution, and its value indicates how that feature is expressed. This representation is analogous to genes in biological organisms, where each gene contributes independently to the overall phenotype.
2. **Selection:** Candidate solutions with less cost are given greater probability of being chosen for reproduction, reflecting the principle of survival of the fittest.
3. **Crossover (Recombination):** The primary reproduction operator in classical GAs is *bit-string crossover*, where two parent strings exchange subsequences of their genetic material to form new offspring. This promotes the inheritance of desirable traits from both parents.
4. **Mutation:** Another essential operator is *bit-flipping mutation*, in which a single bit of a chromosome is inverted ($0 \rightarrow 1$ or $1 \rightarrow 0$). This operator maintains genetic diversity and prevents premature convergence by exploring new regions of the search space.
5. **Replacement:** Newly generated offspring replace less-fit individuals in the population, ensuring the propagation of stronger solutions across generations.

3.3. The CFE approximation

This technique serves as the **main mathematical framework for approximating the fractional Laplace operator** through **integer-order rational transfer functions**. The method is established based on the following approximation: [15, 24, 25]:

$$(1+z)^\alpha = \frac{1}{1 - \frac{\alpha z}{1 + \frac{(1+\alpha)z}{2 + \frac{(1-\alpha)z}{3 + \frac{(2+\alpha)z}{2 + \frac{(2-\alpha)z}{3 + \frac{(n-\alpha)z}{2 + \frac{(n-\alpha)z}{2 + \frac{(n-\alpha)z}{2 + \frac{(n-\alpha)z}{2 + \dots}}}}}}}}}}, \quad (7)$$

where $0 < \alpha < 1$ and $n \in \mathbb{N}$. For the purpose of obtaining a finite-order approximation of the operator s^α , one might replace the term s instead of the variable z in (7). This exchange step enables the n^{th} -order approximation of such an operator to appear around the center frequency $\omega_0 = 1$ rad/sec as follows [18, 19]:

$$s^\alpha \cong \frac{\alpha_0 s^n + \alpha_1 s^{n-1} + \dots + \alpha_{n-1} s + \alpha_n}{\alpha_n s^n + \alpha_{n-1} s^{n-1} + \dots + \alpha_1 s + \alpha_0}, \quad (8)$$

where $0 < \alpha_i < 1$, $i = 0, 1, 2, \dots, 5$. Moreover, the operator $s^{-\alpha}$ can be simply obtained by inverting upside-down the expression given in (8).

3.4. Oustaloup's approximation

Oustaloup's approximation is a widely adopted technique for constructing integer-order transfer functions that emulate fractional-order operators s^α [21, 26] (where $0 < \alpha < 1$). This method works by matching the frequency response of s^α over a specified frequency range $[\omega_l, \omega_h]$ using a rational transfer function consisting of strategically

placed poles and zeros. The approximation can be expressed as:

$$s^\alpha \approx \prod_{k=-N}^N \frac{s + \omega'_k}{s + \omega_k} = \frac{B_n s^n + B_{n-1} s^{n-1} + \dots + B_1 s + B_0}{A_n s^n + A_{n-1} s^{n-1} + \dots + A_1 s + A_0}, \quad (9)$$

where:

- N is the order of the approximation,
- ω_k and ω'_k are the poles and zeros, respectively,
- K is a gain adjustment factor ensuring correct magnitude at a reference frequency.
- The unity-gain geometric frequency ω_u is calculated from the following:
 $\omega_u = \sqrt{\omega_b \omega_h}$.

The poles and zeros are logarithmically spaced and computed as:

$$\omega_k = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{k+N+0.5(1+\alpha)}{2N+1}}, \quad \omega'_k = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{K+N+0.5(1-\alpha)}{2N+1}}, \quad K = \left(\frac{\omega_h}{\omega_b} \right)^{-\frac{\alpha}{2}} \prod_{k=-N}^N \frac{\omega_k}{\omega'_k}$$

The accuracy of the approximation depends on the filter order N and the selected frequency range (ω_b, ω_h) . Note that the resulting transfer function in equation (9) always has an order of $n = 2N + 1$. In the special case where the frequency limits ω_b and ω_h are symmetric around the central frequency $u = 1$ rad/s (i.e., $\omega_b = 1/\omega_h$), the coefficients of equation (9) become interrelated as follows:

$$A_{i-1} = B_i, \quad i = 0, 1, 2, 3, \dots, N.$$

4. Experimental Setup

4.1. Optimization Range

The choice of PID controller values was very tricky since the plant poles are very close to the imaginary axis. The VCCS plant simplified transfer function was shown in equation 3. The root locus of this is shown in figure 3.

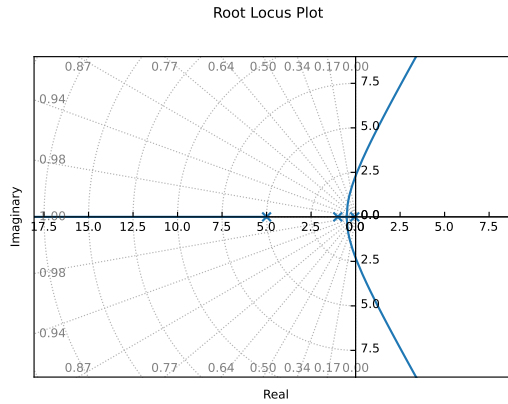


Figure 3. Root Locus of Plant Transfer Function

So, we have to be careful with the bounds of K_p , K_i , and K_d since large values can lead to system instability. The breakout to right hand side (RHS) occurred at $K_p \approx 12.81$. So, the bound of K_p in the GA was limited to be $[0.1, 10]$. This was arbitrarily selected since using K_i and K_d may reduce the stability margin less than 12.81. Arbitrary K_i bounds and K_d bounds were set to $[0, 5]$ and $[0, 3]$ respectively. Both of the bounds of γ and ρ were set to $[0, 1]$

4.2. GA Setting

Genetic algorithm population of 10 chromosomes. Each chromosome contains 5 genes: K_p , K_i and K_d , γ , ρ (only 3 genes are used in the case of classical PID). These genes were first randomly generated, and the bounds chosen above were enforced on these values to minimize unstable systems. The GA algorithm were run for 30 generations. During each generation the bottom 5 chromosomes from the population were selected, i.e, the 5 chromosomes with the lowest cost function. These selected chromosomes were randomly crossover to generate the next generation. A single crossover operation is done between two chromosomes that were randomly selected from the chosen ones. The genes of these chromosomes were combined with some random factor between $[0, 1]$. Also, Mutation was randomly applied on 10% of the resulted children with bounds between $[0.05 \ 0.1]$. No termination was enforced until the algorithm completes all generations, and the best results were recorded for each generation. The cross over was performed between the population that was selected such that they have the lowest const function. Since 5 population The above GA process was repeated for 10 times to explore more results and the best result were recorded.

4.3. FOPID Number of Terms

In this research the Oustalop and the CFE approximation will be highly affected by the considered number of terms. So N is set for 2, and 3. best results appears when $N=2$. So, the number of terms considered are $2N+1 = 5$ for each fractional order.

4.4. Cost Function

Within the optimization framework, a cost function is formulated such that minimizing its value corresponds to achieving optimal controller performance. This cost function incorporates four closed-loop performance criteria—steady-state error, settling time, rise time, and maximum overshoot. It is mathematically expressed as:

$$J = f(E_{ss}, T_s, T_r, M_p)$$

where E_{ss} is the steady-state error, T_s is the settling time, T_r is the rise time, and M_p represents the maximum overshoot. [16]: Selecting standard cost functions such as the ITAE (Integral Time Absolute Error) or the MSE (mean square error) results in high overshoot (around 40% in most cases) which lead to unacceptable behavior, especially in the VCCS application where speed limit is enforced by law, and also high overshoot implies safety hazards. Moreover, and the response time and settling time has more tolerance. So, it is acceptable in the VCCS application to have a rise time between 5 to 10 seconds, rather than less than a second in the case of using ITAE. Because of this, a custom cost function was set to be more appropriate to the application in hand as shown in equation 10.

$$V = w_1 E_{ss} + w_2 M_p + w_3 T_r + w_4 T_s, \quad (10)$$

where w_1, w_2, w_3 and w_4 are weights that are selected to affect the importance of the listed performance criteria in the cost function. The value of the weights are $w_1 = 10.0, w_2 = 10.0, w_3 = 1.0$ and $w_4 = 1.0$. High penalty was set on maximum overshoot to avoid over speeding. Steady state error has also the same high weight. Low penalty was set on rise time and settling time. This cost function was selected such that the controller should emphasize on avoiding high overshoot and maintaining the intended speed with minimum steady state error.

5. Results and Discussion

Genetic Algorithms (GAs) are employed as optimization tools to search for feasible solutions when tuning controller parameters. Although GAs do not guarantee finding the global optimum, they can produce sufficiently good or acceptable solutions within available computational resources, such as time and memory. As heuristic search methods, GAs trade strict optimality guarantees for flexibility, making them highly effective in scenarios where feasibility and robustness are prioritized over absolute optimality.

In this work, the search space boundaries are defined for practical implementation. Specifically, the optimization considers the PID controller parameters K_p , K_i , and K_d , as well as the fractional integral and derivative orders γ and ρ . The bounds for these parameters are chosen as follows:

$$K_p \in (0, 20], \quad K_i \in (0, 5], \quad K_d \in (0, 3], \quad \gamma \in (0, 1], \quad \rho \in (0, 1].$$

The GA was repeated 10 times on each configuration: PID, FOPID with CFE approximation and FOPID with Oustaloup approximation.

5.1. Conventional PID Results

For the classical PID controller, where $\gamma = \rho = 1$, the Genetic Algorithm (GA) was executed 10 times. The best results obtained from these runs are summarized as follows:

Optimized Parameters

$$K_p = 3.21, \quad K_i = 0.67, \quad K_d = 3, \\ \gamma = 1 \quad (\text{integral order}), \quad \rho = 1 \quad (\text{derivative order}).$$

Performance Metrics

$$\begin{aligned} \text{Steady-State Error } (E_{ss}) &= 0, \\ \text{Maximum Overshoot } (M_p) &= 8.77\%, \\ \text{Settling Time } (T_s) &= 1.620\text{s}, \\ \text{Rise Time } (T_r) &= 0.96\text{s}, \\ \text{Cost Value } (V) &= 90.316. \end{aligned}$$

Hence, the transfer function for the classical PID controller is given by equation 11:

$$C(s) = 3.211 + 3s + \frac{0.6691}{s} \quad (11)$$

and the closed-loop transfer function is given by equation 12:

$$H(s) = \frac{7.425s^6 + 52.86s^5 + 89s^4 + 53.83s^3 + 10.65s^2 + 0.3944s}{s^8 + 12.1s^7 + 54.57s^6 + 117.3s^5 + 119.8s^4 + 56.34s^3 + 10.71s^2 + 0.3944s} \quad (12)$$

5.2. FOPID with CFE Approximation Results

Regarding the FOPID controller, repeating the GA 10 times with the CFE approximation produced the following optimal values:

Optimized Parameters

$$K_p = 5.264943, \quad K_i = 1.98399374, \quad K_d = 2.52803494, \\ \gamma = 0.9999 \quad (\text{integral order}), \quad \rho = 0.9999 \quad (\text{derivative order}).$$

Performance Metrics

$$\begin{aligned} \text{Steady-State Error } (E_{ss}) &= 0.016, \\ \text{Maximum Overshoot } (M_p) &= 16.97\%, \\ \text{Settling Time } (T_s) &= 7.2\text{s}, \\ \text{Rise Time } (T_r) &= 0.6\text{s}, \\ \text{Cost Value } (V) &= 177.66 \end{aligned}$$

Controller Transfer Function The resulting FOPID controller using CFE realization is shown in equation 13:

$$C_1(s) = \frac{17.29s^4 + 57.08s^3 + 99.15s^2 + 87.61s + 29.81}{s^4 + 6.249s^3 + 15.95s^2 + 15.52s + 5.048} \quad (13)$$

Closed-Loop System Numerator of the closed-loop transfer function $H_2(s) = \frac{C_2(s)G(s)}{1+C_2(s)G(s)}$ is shown in equation 14

$$CFEClosedNum = 42.81s^{11} + 667.8s^{10} + 4511s^9 + 1.78e04s^8 + 4.568e04s^7 + 8.048e04s^6 + 9.898e04s^5 + 8.372e04s^4 + 4.651e04s^3 + 1.548e04s^2 + 2503s + 88.7 \quad (14)$$

The denominator of the CFE closed-loop transfer function is shown in equation 15

$$CFEClosedDen = s^{14} + 24.59s^{13} + 269.3s^{12} + 1785s^{11} + 8094s^{10} + 2.644e04s^9 + 6.363e04s^8 + 1.138e04s^7 + 1.52e05s^6 + 1.509e05s^5 + 1.088e05s^4 + 5.402e04s^3 + 1.668e04s^2 + 2576s + 90.14 \quad (15)$$

5.3. FOPID with Oustaloup Approximation Results

For the Oustaloup's approximation, with frequency limits $(\omega_{low}, \omega_{high})$ defined as [0.01, 100], the most optimal results are presented below in terms of parameters and performance metrics:

Optimized Parameters

$$K_p = 5.264943, \quad K_i = 1.98399374, \quad K_d = 2.52803494, \\ \gamma = 0.9999 \quad (\text{integral order}), \quad \rho = 0.9999 \quad (\text{derivative order}).$$

Performance Metrics

$$\begin{aligned} \text{Steady-State Error } (E_{ss}) &= 8.1 \times 10^{-4}, \\ \text{Maximum Overshoot } (M_p) &= 16.74\%, \\ \text{Settling Time } (T_s) &= 4.32s, \\ \text{Rise Time } (T_r) &= 0.9s, \\ \text{Cost Value } (V) &= 172.62. \end{aligned}$$

Controller Transfer Function The fractional order controller numerator of Oustaloup's approximation is polynomial of the 10 degree shown in equation 16:

$$OustCtlNum = [2532, 3.463e5, 1.356e7, 1.192e8, 4.083e8, 5.942e8, 3.668e8, 9.944e7, 1.08e7, 2.731e5, 1988] \quad (16)$$

and the denominator is also a polynomial of the 10 degree shown in equation 17:

$$OustCtlDen = [1, 1135, 1.397e5, 5.109e6, 3.646e7, 8.155e7, 3.646e7, 5.109e6, 1.397e5, 1135, 1] \quad (17)$$

Oustaloup Closed-Loop System The numerator of the closed-loop transfer function $H_2(s) = \frac{C_2(s)G(s)}{1+C_2(s)G(s)}$ is shown in equation 18:

$$\begin{aligned} OustClosedNum = [6.119e3, 8.233e6, 2.4e9, 3.04e11, \\ 1.956e13, 6.54e14, 1.118e16, 1.087e17, \\ 6.430e17, 2.404e18, 5.739e18, 8.736e18, \\ 8.453e18, 5.177e18, 1.993e18, 4.757e17, \\ 6.873e16, 5.831e15, 2.859e14, 8.049e12, \\ 1.280e11, 1.087e9, 4.197e6, 4.258e3] \end{aligned} \quad (18)$$

The denominator of the closed-loop transfer function is shown in equation 19:

$$\begin{aligned} OustClosedNum = [1, 2.35e3, 1.748e06, 4.496e8, 5.486e10, 3.539e12, 1.236e14, \\ 2.340e15, 2.646e16, 1.919e17, 9.332e17, 3.122e18, 7.295e18, 1.2e19, \\ 1.39e19, 1.119e19, 6.103e18, 2.191e18, 5.018e17, 7.08e16, 5.932e15, \\ 2.89e14, 8.103e12, 1.286e11, 1.09e9, 4.202e6, 4.262e3] \end{aligned} \quad (19)$$

5.4. Comparative Analysis

To illustrate the performance differences between these techniques, the numerical results of the step response of $H_1(s)$ (CFE-based) and $H_2(s)$ (Oustaloup-based) are compared with the classical PID $H_3(s)$. Figures 4–6 show the step responses of the three controllers, representing their transient and steady-state behavior under identical input conditions.

- Classical PID Response

Figure 4 illustrates the step response of the classical PID controller. The rise time T_r is the slowest, however, the difference is not significant, and the settling time is the shortest since it does not show any oscillation. Also, the overshoot is as half as the other FOPID methods. The system stabilizes relatively quickly. The PID provides a good balance between speed and stability. This explains its widespread use in industrial control systems, where simplicity, low overshoot and quick response are often prioritized.

- FOPID-CFE Response

As shown in Figure 5, the FOPID-CFE controller exhibits the fastest rise time T_r . However, it produces a noticeable overshoot (around 17%), slow settling time T_s and the largest steady state error 1.6%. While this configuration enhances responsiveness by reducing rise time, it does not compete with the other two methods in terms of steady state error.

- FOPID–Oustaloup Response

Figure 6 shows an improved transient behavior for the Oustaloup-based FOPID controller. The rise time is the very close to PID. The overshoot is the very close to the CFE method. The response converges smoothly to the steady-state value without oscillations. Also, despite the existence of steady state error 8.1×10^{-4} , the Oustaloup approximation ensures a better damped and stable dynamic performance than the FOPID-CFE. The classical PID shows the best result in the step response among other three methods.

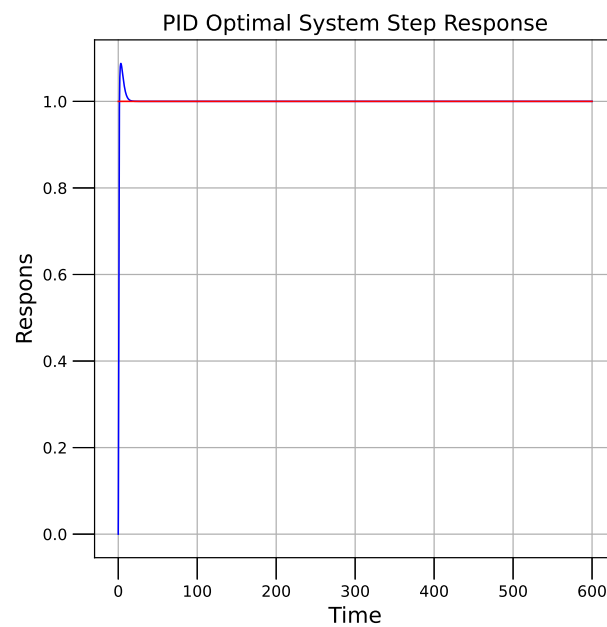


Figure 4. Step responses of Classical PID step response

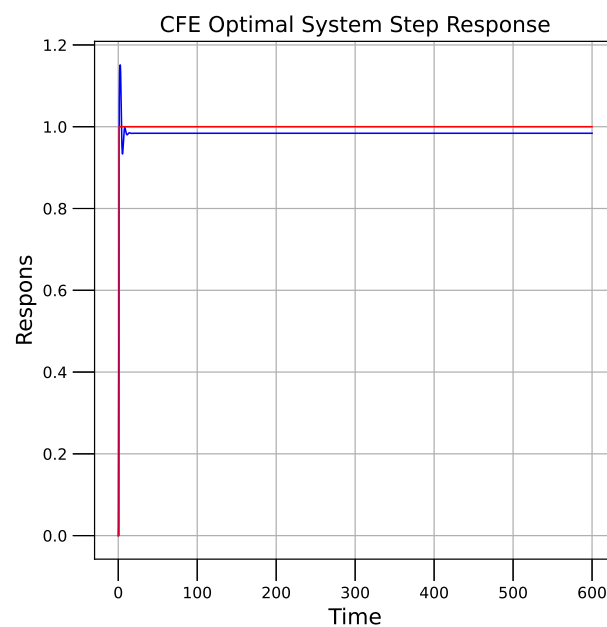


Figure 5. FOPID step response using CFE

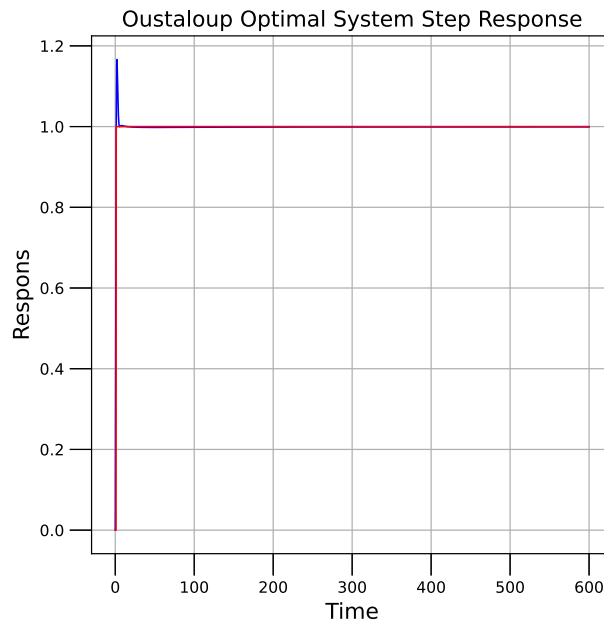


Figure 6. FOPID step response using Oustaloup

Table 1. Comparison of step response characteristics

Parameter	PID	FOPID-CFE	FOPID-Oust
Rise Time (s)	0.96s	0.6s	0.9s
Settling Time (s)	1.62s	4.14s	4.32s
Maximum Overshoot (%)	8.77	16.97	16.74
Steady-State Error	0	0.016	8.1×10^{-4}
Cost Function	90.316	177.66	172.62
Cost std	126.95	110.93	181.32
Cost Confidence Interval	[273.19, 284.98]	[323.25, 332.21]	[428.51, 443.17]

5.5. Disturbance Analysis

In order to test the disturbance rejection of the resulted optimized systems, an input disturbance signal is injected in the system after the controller block and before the plant block via a summing junction. This input disturbance might represent high speed wind or road slope. The disturbance signal was set to be 600 seconds long with one main disturbance jumps from 0 to 1 (step) starts at second 100 and continue to the end of the signal. During the disturbance testing, the input was set to zero. In case the input was not zero, and considering an LTI system, the disturbance response should be superposed with the input response. The block diagram of the system including disturbance signal is shown in figure 7.

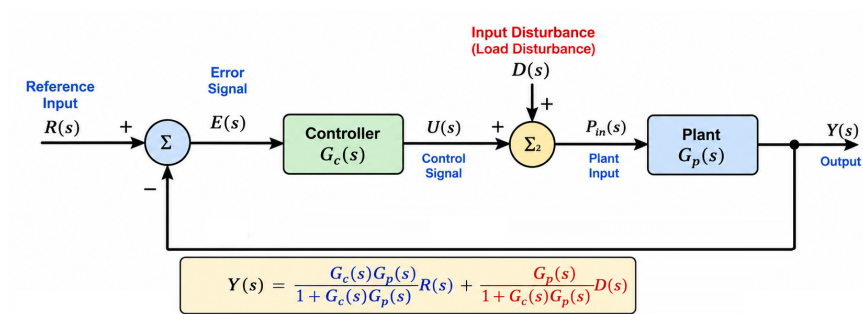


Figure 7. Block diagram depicts the input disturbance signal

Figures 8, 9 10 show the disturbance response for the system using the classical PID, FOPID CFE controller and FOPID Oustaloup controller respectively. The classical PID shows the best disturbance rejection behaviour in terms of recovery time and acceptable peak disturbance error. The FOPID CFE system shows worst steady-state regulation error. The FOPID Oustaloup shows acceptable behaviour, although it is far from the classical PID. So, again the classical PID is clearly the winner in the disturbance rejection response.

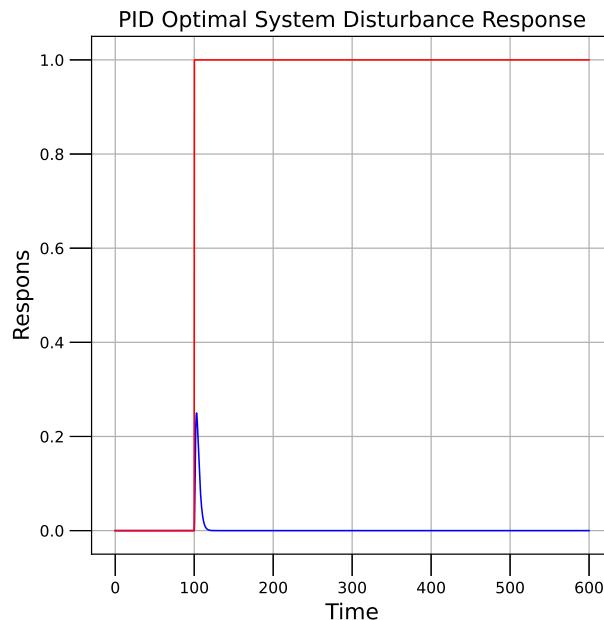


Figure 8. Classical PID disturbance rejection response

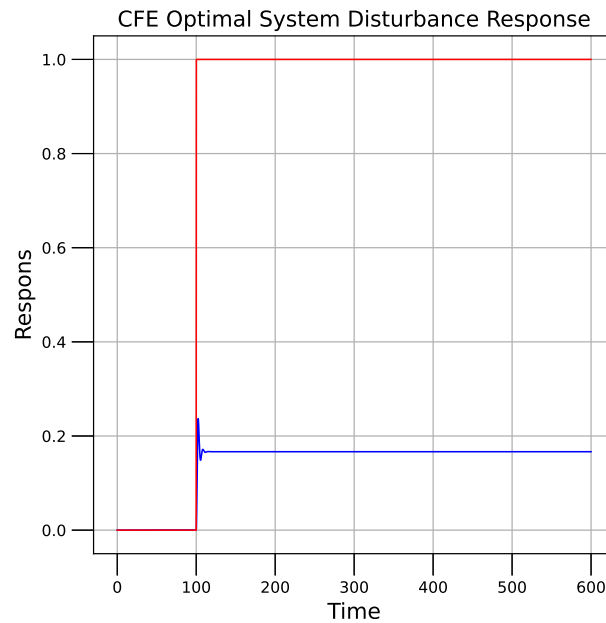


Figure 9. FOPID disturbance rejection response using CFE

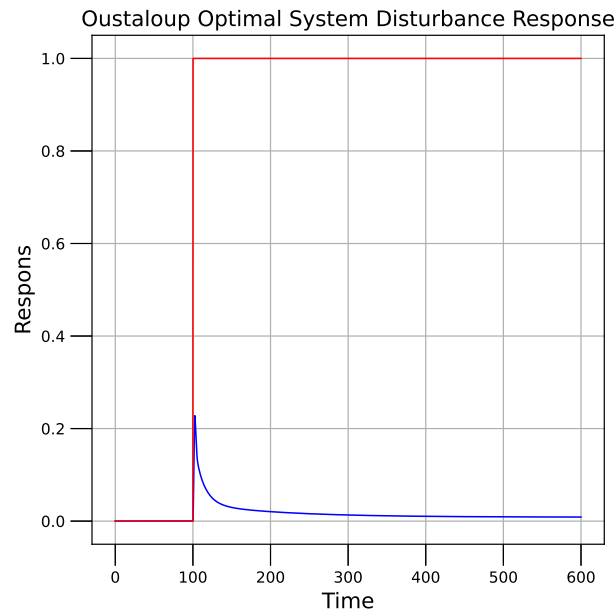


Figure 10. FOPID disturbance rejection response using Oustaloup

Table 2 shows the disturbance response characteristics of the three systems. As can be seen from the result, the classical PID show better results than both the FOPID. Also it is worth mentioning the the steady state error in the FOPID CFE case remains high.

Table 2. Comparison of Disturbance Rejection Response Characteristics

Parameter	PID	FOPID-CFE	FOPID-Oust
Recovery Time (s)	0.133 s	4.35	448.69
Maximum Deviation (%)	24.99	23.69	22.79
Steady-State Error	0	.167	8.5×10^{-3}

5.6. Open Loop Bode Analysis

Bode plots for the open loop transfer function for the classical PID, FOPID-CFE and FOPID-Oustaloup are shown in figure , and 13 respectively. Figure 11 presents the open-loop Bode plot of the system with the classical PID controller. The magnitude response exhibits a monotonic decrease as the frequency increases, indicating effective attenuation of high-frequency components and good noise rejection capability. At low frequencies, the gain remains relatively high, suggesting satisfactory tracking performance and disturbance rejection in the low-frequency region. The phase response decreases smoothly from approximately 0° toward -270° , reflecting the cumulative phase lag introduced by the controller and plant dynamics. No abrupt changes or resonance peaks are observed in either the magnitude or phase plots, indicating a well-behaved frequency response and the absence of lightly damped modes. Overall, the Bode characteristics suggest that the classical PID controller provides a stable frequency response with adequate low-frequency gain and progressive attenuation of high-frequency signals, although the increasing phase lag at higher frequencies may reduce the phase margin and limit the achievable closed-loop bandwidth.

Figure 12 illustrates the open-loop Bode plot of the system employing the CFE-based controller. The magnitude response exhibits a gradual decrease with increasing frequency, indicating strong attenuation of high-frequency components while maintaining relatively high gain at low frequencies for effective reference tracking and low-frequency disturbance rejection. Unlike the classical PID controller, the magnitude response shows a slight flattening in the mid-frequency range, suggesting that the CFE approximation modifies the frequency characteristics by introducing additional dynamics. The phase response decreases smoothly from approximately 0° toward -270° , with a noticeable phase lead (or reduced phase lag) in the intermediate frequency region before continuing its downward trend. This may contribute to enhanced phase margin and improved transient performance. Furthermore, in the magnitude response, the absence of resonance peaks indicates stable open-loop behavior.

Figure 13 demonstrates a stable and well-behaved frequency response over a wide frequency range. At low frequencies, the magnitude is relatively high, indicating strong low-frequency gain and good capability for tracking low-frequency reference signals. As the frequency increases, the magnitude decreases continuously with no resonant peaks, suggesting the absence of oscillatory behavior and indicating a stable open-loop characteristic. Same as the other two, The phase response exhibits a smooth and progressive decrease from approximately 0° toward -270° at high frequencies. Overall, the frequency response indicates that the Oustaloup-based model maintain a stable and predictable open-loop response.

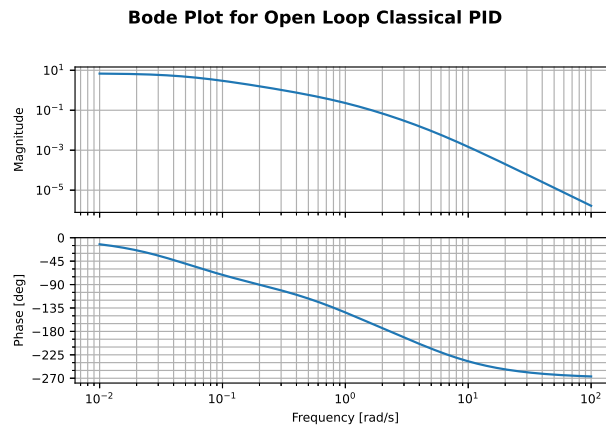


Figure 11. Bode Plot for Open Loop PID

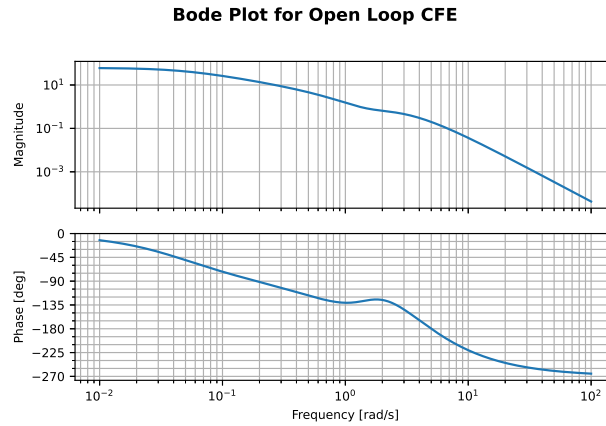


Figure 12. Bode Plot for Open Loop FOPID CFE

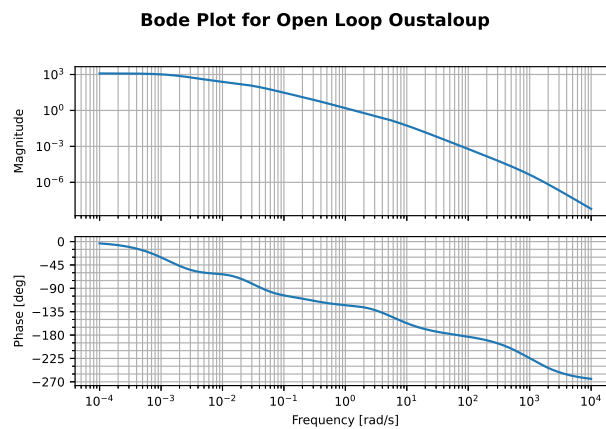


Figure 13. Bode Plot for Open Loop FOPID Oust

6. Conclusion

The classical PID has the best performance in terms of minimum overshoot and near zero steady state and acceptable rise time. Also, in the disturbance rejection response, classical PID has better performance in terms of low maximum deviation and low recovery time. Theoretically, the FOPID should perform at least as better as the classical PID if the used optimization method guarantee to have optimal solution. However, since the GA optimization method does not guarantee to have the optimal solution, the result where the PID shows better performance than the FOPID using the CFE realization.

REFERENCES

1. T. Hamadneh, O. Alsayyed, B. Batiha, E. M. Barhoumi, M. A. Kamarposhti, H. Haghighi, P. Thounthong, and I. Colak, *Optimal energy management of distributed generation resources in a microgrid under various load and solar irradiance conditions using the artificial bee colony algorithm*, Scientific Reports, vol. 15, no. 1, Art. 31097, 2025.
2. Z. Chen, S. Xiong, Q. Chen, Y. Zhang, J. Yu, J. Jiang, and C. Wu, *Eco-driving: A scientometric and bibliometric analysis*, IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 22716–22736, 2022.
3. D. Izci and S. Ekinici, *A novel hybrid ASO-NM algorithm and its application to automobile cruise control system*, Proceedings of the 2nd International Conference on Artificial Intelligence: Advances and Applications (ICAIAA 2021), pp. 333–343, 2022.
4. K. Osman, M. F. Rahmat, and M. A. Ahmad, *Modelling and controller design for a cruise control system*, Proceedings of the 5th International Colloquium on Signal Processing & Its Applications, pp. 254–258, 2009.
5. D. Hu, J. Zhao, J. Zheng, and H. Liu, *An adaptive cruise control strategy for intelligent vehicles based on hierarchical control*, World Electric Vehicle Journal, vol. 15, no. 11, Art. 529, 2024.
6. R. Jain, S. Agrawal, and R. Sharma, *Tuning of digital PID controller for blood glucose level of diabetic patient*, International Research Journal of Engineering and Technology, vol. 4, pp. 499–504, 2017.
7. P. Dua, F. J. Doyle III, and E. N. Pistikopoulos, *Model-based blood glucose control for type 1 diabetes via parametric programming*, IEEE Transactions on Biomedical Engineering, vol. 53, no. 8, pp. 1478–1491, 2006.
8. S. Momani, I. M. Batiha, and R. El-Khazali, *Design of $PI^\lambda D^\delta$ heart rate controllers for cardiac pacemaker*, Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 1–5, 2019.
9. L. Abualigah, S. Ekinici, D. Izci, and R. Zitar, *Modified elite opposition-based artificial hummingbird algorithm for designing FOPID controlled cruise control system*, Intelligent Automation and Soft Computing, vol. 38, no. 2, 2023.
10. I. M. Batiha, R. El-Khazali, and S. Momani, *Dynamic responses of PI^λ , PD^δ , and $PI^\lambda D^\delta$ controllers for prosthetic hand model using PSO algorithm*, Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 1–6, 2019.
11. S. Momani, R. El-Khazali, and I. M. Batiha, *Tuning PID and $PI^\lambda D^\delta$ controllers using particle swarm optimization algorithm via El-Khazali's approach*, AIP Conference Proceedings, vol. 2172, no. 1, Art. 050003, 2019.
12. R. El-Khazali, I. M. Batiha, and S. Momani, *Approximation of fractional-order operators*, in Fractional Calculus, Springer Proceedings in Mathematics & Statistics, vol. 303, pp. 121–151, Springer, 2019.
13. R. El-Khazali, S. Momani, and I. M. Batiha, *A new discrete-time model of fractional-order PLL*, Proceedings of the 3rd International Conference on Signal Processing and Information Security (ICSPIS), pp. 1–4, 2020.
14. I. Podlubny, L. Dorčák, and I. Kostial, *On fractional derivatives, fractional-order dynamic systems and $PI^\lambda D^\mu$ controllers*, Proceedings of the 36th IEEE Conference on Decision and Control, vol. 5, pp. 4985–4990, 1997.
15. B. T. Krishna, *Studies on fractional-order differentiators and integrators: A survey*, Signal Processing, vol. 91, no. 3, pp. 386–426, 2011.
16. S. Momani and I. M. Batiha, *Tuning of the fractional-order PID controller for some real-life industrial processes using particle swarm optimization*, Progress in Fractional Differentiation and Applications, vol. 8, no. 4, pp. 377–391, 2022.
17. R. M. Batyha, *Optimal design of fractional-order PID controllers using bacterial foraging optimization algorithm*, International Journal of Advances in Soft Computing and Its Applications, vol. 13, no. 1, pp. 136–149, 2021.
18. I. M. Batiha, S. A. Njadat, R. M. Batyha, A. Zraiqat, A. Dababneh, and S. Momani, *Design fractional-order PID controllers for single-joint robot arm model*, International Journal of Advances in Soft Computing and Its Applications, vol. 14, pp. 96–114, 2022.
19. I. M. Batiha, R. B. Albadarneh, S. Momani, and I. H. Jebril, *Dynamics analysis of fractional-order Hopfield neural networks*, International Journal of Biomathematics, vol. 13, no. 8, Art. 2050083, 2020.
20. A. Idir, A. Zemmit, K. Khettab, M. Nesri, S. Guedida, and L. Canale, *Performance analysis of fractional-order PID controller based on metaheuristic optimisation algorithms for vehicle cruise control systems*, Science, Engineering and Technology, vol. 5, no. 1, pp. 1–14, 2025.
21. S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*, Springer, Berlin, Germany, 2007.
22. M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, USA, 1998.
23. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, USA, 1989.
24. E. Piotrowska and R. Melnik, *Analysis of fractional electrical circuit containing two RC ladder elements of different fractional orders*, Acta Mechanica et Automatica, vol. 18, no. 1, pp. 77–83, 2024.
25. A. D. Sokal, *A simple algorithm for expanding a power series as a continued fraction*, Expositiones Mathematicae, vol. 41, no. 2, pp. 245–287, 2023.
26. J. Wiora and A. Wiora, *Influence of methods approximating fractional-order differentiation on the output signal illustrated by three variants of Oustaloup filter*, Symmetry, vol. 12, no. 11, Art. 1898, 2020.