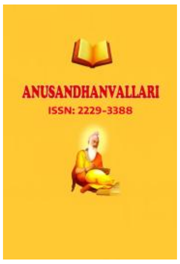

Anusandhanvallari
Vol 2023, No.1
November 2023
ISSN 2229-3388

Fractional-Order PID Controller Design for a Transfer-Function-Based Third-Order System Using Particle Swarm Optimization

Niranjan Behera & Mihira Kumar Nath

Department of Electrical Engineering GCE Kalahandi, Bhawanipatna, Odisha

niranjan.265@gmail.com, mihirnath.83@gmail.com

ABSTRACT

In recent times, the application of fractional derivatives has become quite apparent in modeling mechanical and electrical properties of real materials. Fractional integrals and derivatives have found wide application in the control of dynamical systems, when the controlled system or/and the controller is described by a set of fractional order differential equations. This paper presents an effective tuning strategy for conventional and fractional-order PID controllers applied to the load frequency control problem of a multi-area power system. The proposed approach combines classical tuning methods with modern optimization techniques to enhance closed-loop dynamic performance. Initial controller parameters are obtained using the Ziegler–Nichols and Astrom–Hagglund methods and further refined using Particle Swarm Optimization to minimize the Integral Squared Error performance index. A state-space model of a four-area interconnected power system is developed, and system responses are evaluated through MATLAB/Simulink simulations. Comparative analysis is carried out among conventional PID, PSO-tuned PID, and PSO-tuned fractional-order PID controllers. Simulation results demonstrate that the PSO-tuned fractional-order PID controller provides superior transient performance, including reduced overshoot, faster settling time, and improved stability margins. The study confirms that combining fractional-order control with intelligent optimization techniques significantly enhances load frequency control performance in multi-area power systems.

Key Words: PID controller, fractional order control, optimization, control systems

1 Introduction:

Proportional–Integral–Derivative (PID) controllers are the most widely used control strategies in industrial applications due to their simple structure, ease of implementation, and satisfactory performance over a wide range of operating conditions. Since their early development, PID controllers have played a crucial role in regulating process variables such as temperature, speed, pressure, and voltage in both linear and nonlinear systems. One of the earliest and most influential contributions to PID controller tuning was made by Ziegler and Nichols, who proposed practical tuning rules based on system step response and ultimate gain characteristics, which are still widely used as a baseline method in controller design [1]. Despite their popularity, conventional PID controllers with integer-order integration and differentiation may not always provide optimal performance, especially for complex and higher-order systems. To address these limitations, several researchers have proposed advanced tuning techniques and controller structures. Astrom and co-workers introduced self-tuning and auto-tuning PID controllers, significantly improving adaptability and robustness in practical applications [2], [3] & [4]. Further developments in automatic and optimal tuning methods have demonstrated that appropriate parameter selection plays a critical role in achieving desired transient and steady-state performance [5].

Fractional-order calculus has emerged as a powerful mathematical tool that extends classical integer-order differentiation and integration to non-integer orders. Manabe first highlighted the applicability of non-integer integrals in control systems, opening new di- ructions for controller design [6]. Based on this concept, fractional-order PID controllers, commonly known as $PI^{\lambda}D^{\mu}$ controllers, introduce additional degrees of freedom through fractional integration and differentiation orders. These controllers offer enhanced flexibility and improved robustness compared to conventional PID controllers [7]. However, the increased number of controller parameters in fractional-order PID controllers makes tuning more challenging. To overcome this issue, optimization-based approaches have been increasingly adopted. Particle Swarm Optimization (PSO), inspired by the social behaviour of biological populations, has proven to be an effective and computationally efficient technique for tuning PID and fractional-order PID controllers [8].

Several studies have demonstrated that PSO-based tuning can significantly enhance control performance in terms of reduced overshoot, faster settling time, and improved robustness [9]. The growing effectiveness of optimization-based techniques for controller tuning has led to extensive research on evolutionary and swarm intelligence algorithms. Shi and Eberhart proposed an adaptive form of Particle Swarm Optimization by integrating fuzzy logic to dynamically adjust the algorithm parameters, which resulted in improved convergence behaviour and reduced chances of premature stagnation [10]. Their work demonstrated that PSO is well suited for solving complex optimization problems commonly encountered in control system design.

Several studies have investigated the application of PSO for tuning PID controllers in mechanical and electromechanical systems. Kao et al. implemented a PSO-based self-tuning PID controller for a slider–crank mechanism and reported enhanced tracking accuracy and improved transient response when compared to conventional tuning techniques [11]. These results confirmed the effectiveness of PSO in handling real-world dynamic systems with nonlinear characteristics.

In addition to PSO, other evolutionary optimization techniques such as genetic algorithms have also been explored for control engineering applications. Wang et al. presented a comprehensive overview of genetic algorithm-based methods and discussed their application to controller parameter tuning problems [12]. Although genetic algorithms possess strong global search capability, their higher computational complexity and slower convergence rates have made PSO a preferred choice in many control applications.

The application of PSO in power system control has also been widely reported. Gaining utilized Particle Swarm Optimization to design an optimal PID controller for an automatic voltage regulator system and demonstrated significant improvements in voltage regulation, transient stability, and robustness [13]. These findings further validated the suitability of PSO-based tuning techniques for power systems requiring fast dynamic response and high reliability.

Based on the reviewed literature, it is evident that PSO has been successfully applied to a wide range of control problems, including mechanical systems, power systems, and nonlinear processes. However, most existing studies primarily focus on the tuning of conventional PID controllers. Comprehensive comparative analyses that extend PSO-based optimization to fractional-order PID controllers and evaluate their performance against both classical and optimized integer-order controllers remain limited. This identified research gap provides the motivation for the present study.

Motivated by these developments, this work focuses on the analysis and tuning of integer-order and fractional-order PID controllers using classical tuning methods and optimization techniques. The performance of conventional PID controllers is compared with that of fractional-order PID controllers, highlighting the advantages of fractional-order control in achieving superior dynamic response and robustness.

This paper is organized to systematically present the design and optimization of a fractional-order PID controller using Particle Swarm Optimization. The focus of the work is derived from the fractional-order controller formulation and optimization framework discussed in Chapter 4 of the project report. Section 1 introduces the motivation for employing fractional-order control and optimization- based tuning techniques. The limitations of conventional integer-order PID controllers and the need for enhanced flexibility in controller design are

Anusandhanvallari
Vol 2023, No.1
November 2023
ISSN 2229-3388

highlighted. Section 2 presents the mathematical formulation of the fractional-order PID (FOPID) controller. The role of fractional integration and differentiation orders is explained, and the relationship between the classical PID controller and the fractional-order controller is established. The advantages of fractional-order control in improving robustness and transient performance are also discussed. Section 3 describes the application of Particle Swarm Optimization for tuning the controller parameters. The PSO algorithm is outlined, and its suitability for multi- parameter optimization is justified. The objective function based on the Integral Squared Error criterion is defined, and the optimization strategy for tuning both integer-order PID and fractional-order PID controllers is detailed. Section 4 explains the system model and simulation setup used for performance evaluation. The simulation environment, optimization parameters, and evaluation criteria are clearly defined to ensure reproducibility of results. Section V presents the simulation results and comparative analysis. The performance of the Ziegler–Nichols tuned PID controller, PSO-tuned PID controller, and PSO-tuned FOPID controller is compared using time-domain specifications such as rise time, settling time, overshoot, and oscillatory behaviour. The effectiveness of the proposed PSO-based fractional-order controller is demonstrated through improved dynamic response. Finally, Section 6 concludes the paper by summarizing the key findings and emphasis-sizing the contribution of fractional-order PID control combined with Particle Swarm Optimization. Possible directions for future research are also briefly discussed.

2 Fractional-Order PID Controller

2.1 Mathematical Representation

Controlling industrial plants requires satisfaction of wide range of specification. So, wide ranges of techniques are needed. Mostly for industrial applications, integer order controllers are used for controlling purpose. Now day's fractional order PID (FOPID) controller is used for industrial application to improve the system control performances. The most common form of a fractional order PID controller is the controller. FOPID controller provides extra degree of freedom for not only the need of design controller gains (K_p, T_i, T_d) but also design orders of integral and derivative. The orders of integral and derivative are not necessarily integer, but any real numbers. The FOPID controller generalizes the conventional integer order PID controller and expands it from point to plane. This expansion could provide much more flexibility in PID control design.

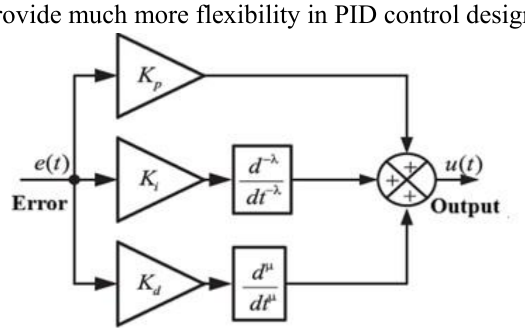

Fig.1 Block Diagram of a Fractional Order PID Controller

The general transfer function of a fractional-order PID controller is expressed as:

$$C(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \quad (1)$$

where K_p, K_i, and K_d denote the proportional, integral, and derivative gains, respectively, while λ and μ represent the fractional orders of integration and differentiation. When $\lambda = 1$ and $\mu = 1$, the controller reduces to a conventional PID controller.

The introduction of fractional orders allows the controller to capture memory and hereditary properties of dynamic systems, which cannot be modelled effectively using integer-order controllers.

2.2 Advantage of fractional order controller: -

As compared to an integer order controller, a fractional order is supposed to offer the following advantages.

• If the parameter of a controlled system changes, a fractional order controller is less sensitive than a classical PID controller.

• FOC (Fractional Order Controller) have two extra variables to tune. This provides extra degrees of freedom to the dynamic properties of fractional order system.

3 Particle Swarm Optimisation (PSO)

3.1 Overview of PSO

Particle Swarm Optimization (PSO), first introduced by Kennedy and Eberhart, is a modern heuristic optimization technique inspired by the social behaviour observed in bird flocking and fish schooling. Owing to its simplicity and efficiency, PSO has been widely recognized as a robust method for solving continuous nonlinear optimization problems.

The fundamental idea of PSO can be explained through a simple scenario in which a group of birds searches for food in each area where only a single food source exists. Although none of the birds initially knows the exact location of the food, each bird can determine how close it is to the food at any given time. The most effective strategy for locating the food is for the birds to follow the individual that is currently closest to the target. This collective behaviour forms the basis of the PSO algorithm for optimization.

In PSO, each candidate solution is represented as a "particle," analogous to a bird moving within the search space. The swarm consists of multiple particles that move through a multidimensional space, each characterized by a position and a velocity. Every particle possesses two key attributes: memory of its own best position (personal best) and awareness of the best position found by the entire swarm (global best). By sharing information about promising positions, particles iteratively update their velocities and positions, guiding the swarm toward the optimal solution.

$$v(k+1)_{i,j} = w.v(k)_{i,j} + c_1 r_1 \left(gbest - x(k)_{i,j}\right) + c_2 r_2 (pbest - x(k)_{i,j}) \quad (2)$$

$$\mathrm{x}(k+1)_{i,j} = x(k)_{i,j} + v(k)_{i,j} \quad (3)$$

Where $V_{i,j}$ represents the velocity of the i_{th} particle in the j_{th} dimension, $X_{i,j}$ denotes the position of the i_{th} particle in the j_{th} dimension, c_1 and c_2 are the acceleration (learning) constants, w is the inertia weight factor, r_1 and r_2 are random numbers uniformly distributed between 0 and 1, p_{best} is the best position achieved by an individual particle, and g_{best} is the best position achieved by the entire swarm.

The PSO algorithm follows a systematic procedure. Initially, a population of particles is generated with randomly assigned positions, velocities, and accelerations. The fitness value of each particle is then evaluated based on the defined objective function. Each particle's current fitness is compared with its previously recorded personal best (p_{best}); if an improvement is observed, p_{best} is updated accordingly. Similarly, the fitness values of all particles are compared to identify and update the global best (g_{best}). Subsequently, the velocity and position of each particle are updated using the predefined velocity and position update equations. These steps are repeated iteratively until a specified stopping criterion, such as maximum iterations or convergence, is satisfied.

Although PSO is a relatively recent optimization technique, it has already been successfully applied to a wide range of engineering and computational problems. One notable application is the optimized particle filter. A particle filter is used to estimate a sequence of hidden states or parameters based solely on observed data, which is particularly useful for noise reduction and improving measurement accuracy. By integrating PSO into the particle filtering process, the estimation performance can be significantly enhanced.

The conventional particle filter often suffers from suboptimal sampling, leading to particle impoverishment. The incorporation of PSO helps mitigate this issue by improving particle diversity and enhancing estimation accuracy. PSO has also been effectively applied to job shop scheduling problems. Job shop scheduling is a complex optimization task in which multiple jobs with varying processing times must be assigned to identical machines while minimizing the overall completion time of the schedule. By reformulating the job shop scheduling problem within the PSO framework, an optimal or near-optimal scheduling solution can be efficiently obtained. Studies have shown that PSO-based solutions are often comparable to or better than the best-known solutions, while being simpler to implement and manipulate than genetic algorithms.

3.2 PSO-Based Tuning of PID and FOPID Controllers

In this study, PSO is employed to optimize the controller parameters by minimizing the Integral Squared Error (ISE), defined as:

$$ISE = \int_0^\infty e^2\,(t)\,dt \tag{4}$$

For the PID controller, the decision variables are K_p, K_i, and K_d. For the FOPID controller, the parameter set is extended to include $\lambda$and μ. Appropriate bounds are assigned to each parameter to ensure practical feasibility.

3.3 Algorithm Implementation

The Particle Swarm Optimization algorithm starts by randomly initializing a population of particles within the predefined bounds of the controller parameters. Each particle represents a potential solution, and its performance is evaluated using the Integral Squared Error as the fitness criterion. Based on this evaluation, the best position achieved by each particle, known as the personal best, and the best solution obtained by the entire swarm, referred to as the global best, are identified and updated iteratively. During each iteration, the velocities and positions of the particles are modified according to their own experience and the collective knowledge of the swarm, allowing the particles to explore and exploit the search space effectively. This iterative optimization process continues until the solution converges to an optimal value or the specified maximum number of iterations is reached.

Figure 2 illustrates the Flow chat of the FOPID controller optimization process using PSO. In this approach, all FOPID controller parameters are updated at each final time instant (t_f). It is important to note that during the optimization process, particle elements may exceed their practical operating limits. To address this issue and based on practical considerations and prior research on FOPID parameter tuning, the lower bounds of all FOPID parameters are set to zero, while suitable upper bounds are predefined.

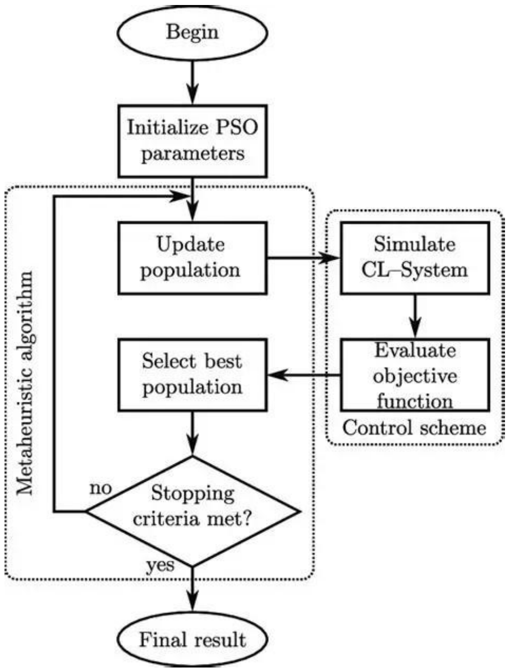

Fig.2 Flow chat for Particle Swam Optimization

4 System Model and Simulation Setup

4.1 Plant model

The control system under consideration is represented by a linear time-invariant (LTI) plant model, which provides a mathematical description of the dynamic behavior of the system. The LTI assumption implies that the system parameters remain constant over time and that the principles of superposition and homogeneity are valid. Such a representation is widely adopted in control system analysis and controller design due to its analytical simplicity and effectiveness. The plant is modeled in the Laplace domain by the following transfer function:

$$G(S) = \frac{1}{s(s+1)(s+5)} \quad (5)$$

The selected plant model captures the essential dynamics required for controller design while maintaining computational efficiency. The control objective is to design an appropriate controller that guarantees closed-loop stability and improves transient response characteristics such as rise time, settling time, overshoot, and steady-state error. This is achieved by shaping the closed-loop poles of the system through suitable controller parameter tuning.

The plant transfer function serves as the foundation for subsequent controller synthesis and performance evaluation. Based on this model, advanced control strategies—such as PID, fractional-order PID (FOPID), or optimization-based controllers—can be systematically designed and analyzed to meet the desired performance criteria.

4.2 Simulation Parameters

MATLAB 2019 is used for simulation and optimization. The PSO parameters include a swarm size of 20 particles, 20 iterations, inertia weight of 0.5, and learning coefficients of 2.4 and 1.6. These values are selected based on convergence performance and computational efficiency as represented in Table-1.

Table-1 Simulation Parameters of the PSO

Parameter	values
Number of iterations	20
Number of trials	5
Swarm size(N)	20
Learning coefficients c1 & c2	2.4 & 1.6
Inertia weight factor w	0.5

5 Results and Discussion

Controller performance is evaluated using standard time-domain specifications, including rise time, settling time, maximum overshoot, and number of oscillations. Several optimization criteria are there for tuning process controllers, such as ISE, IAE, ITSE and ITAE, all of them based on the minimization of integrals of errors. In this case, we propose the minimization of Integral Square of Error (ISE) defined by equation 4. Fig. 3 represents the simulation diagram to get the output using PSO based fractional order PID controller.

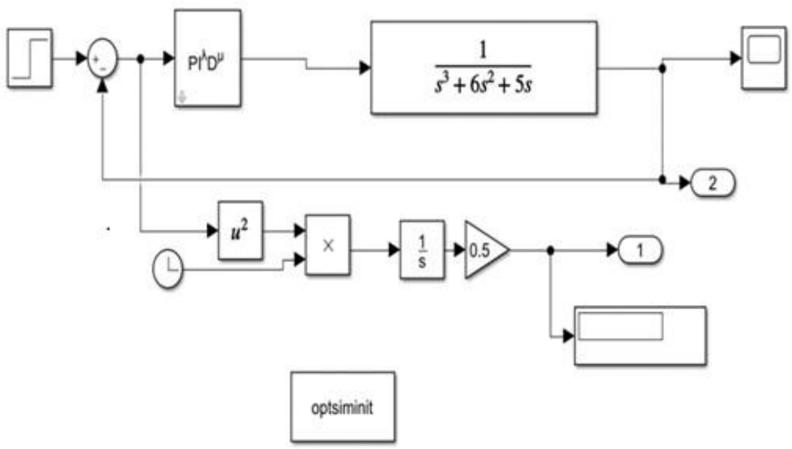

Fig.3 Simulation diagram using PSO

Table-2 Comparison of controller parameters

Controllers	Kp	Ki	Kd	λ	μ
ZNPID	18	12.81	6.32	1	1
PSOPID	30	0.03	16.3091	1	1
PSOFOPID	21.9698	0.01	30	0.9372	1.2136

MATLAB 2019 is used for coding and simulation to tune the proposed PID controllers. Table-2 shows a comparison of ZN PID, PSO based PID and FOPID controller parameters.

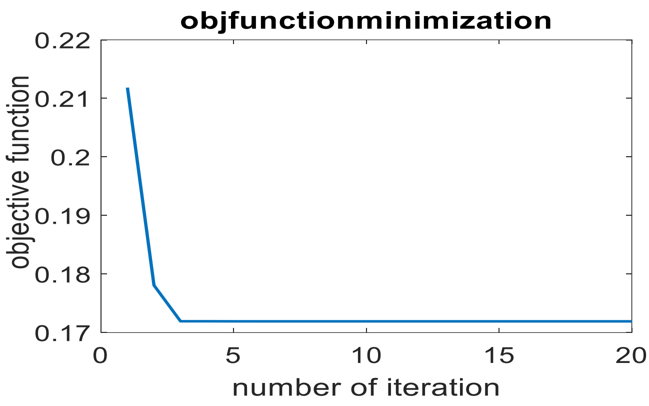

Fig.4 Objective function minimization by PSO to tune PID Controller

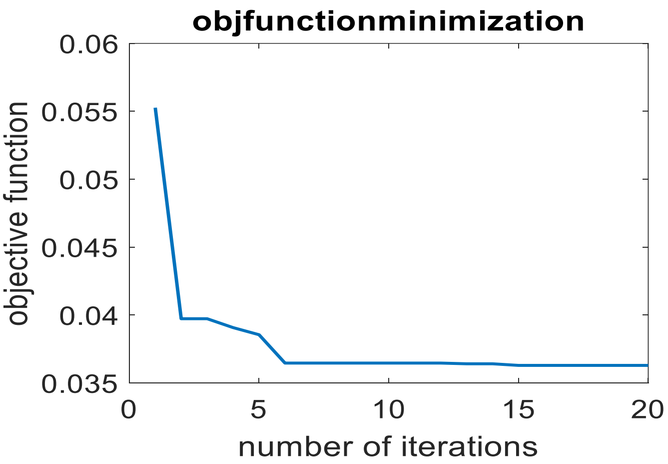

Fig.5 Objective function minimization to tune FOPID using PSO

Figure 4 illustrates the convergence behavior of the Particle Swarm Optimization (PSO) algorithm during the tuning of the conventional PID controller. The plot shows the variation of the objective function value, defined in terms of the Integral Squared Error (ISE), with respect to the number of iterations. As the iterations progress, the objective function value decreases rapidly in the initial stages, indicating fast convergence of the PSO algorithm. After a few iterations, the curve gradually stabilizes, demonstrating that the optimal PID parameters have been reached. This figure confirms the effectiveness of PSO in minimizing the performance index and improving the closed-loop response compared to classical tuning methods.

Figure 5 presents the convergence characteristics of PSO while tuning the fractional-order PID (FOPID) controller parameters. In addition to the proportional, integral, and derivative gains, the fractional integration and differentiation orders are also optimized. Compared to the PID tuning case, the objective function exhibits a smoother and lower final convergence value, indicating improved optimization performance. The reduced ISE value demonstrates that the additional degrees of freedom provided by the fractional orders enable the controller to achieve better tracking accuracy and enhanced dynamic response.

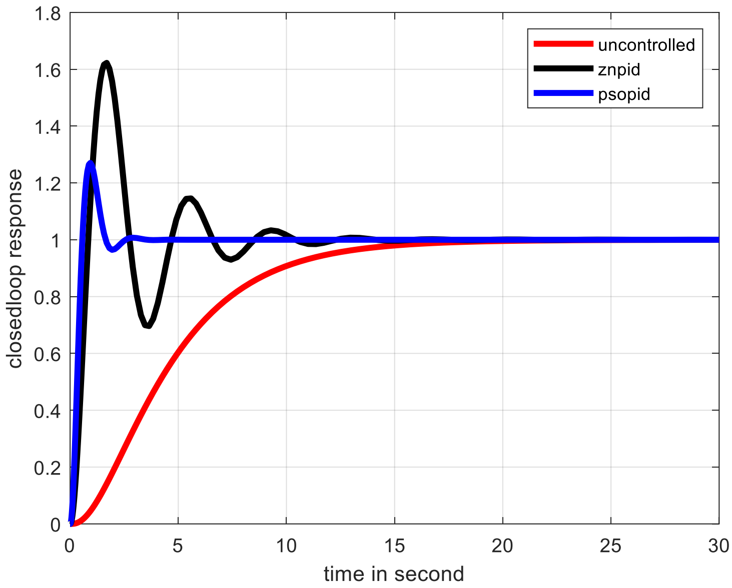

Fig.6 Comparison of Responses of Closed Loop System with ZN PID and PSO PID

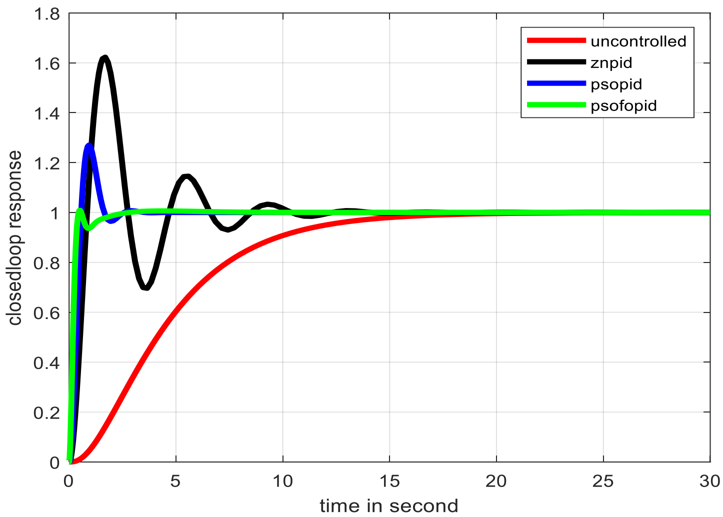

Fig.7 Output response of the closed loop system with ZNPID, PSOPID and PSOFOPID Controllers

Figure 6 compares the time-domain responses of the closed-loop system using a Ziegler–Nichols (ZN) tuned PID controller and a PSO-tuned PID controller. It can be observed that the ZN-PID controller exhibits higher overshoot, longer settling time, and noticeable oscillations. In contrast, the PSO-PID controller significantly improves system performance by reducing overshoot and settling time, and by damping oscillations more effectively. This comparison highlights the superiority of optimization-based tuning over classical heuristic tuning techniques.

Figure 7 presents a comprehensive comparison of the closed-loop output responses obtained using ZN-PID, PSO-PID, and PSO-FOPID controllers. The ZN-PID controller shows the poorest transient performance, characterized by high overshoot and prolonged oscillations. The PSO-PID controller improves the response by reducing oscillations and setting time. However, the PSO-FOPID controller provides the best overall performance, exhibiting minimal overshoot, faster rise time, and the shortest settling time among all controllers. The absence of sustained oscillations in the PSO-FOPID response confirms the enhanced stability and robustness achieved through fractional-order control combined with PSO-based optimization.

Table-3 Comparison of time domain specifications

Controllers	Maximum overshoot (M_p)	Rise time (T_r)	Settling time (T_s)	Number of oscillations
Uncontrolled	1	18	18	0
ZNPID	1.61	0.8673	15	3
PSOPID	1.265	0.5608	3.4635	1
PSOFOPID	1	0.4425	2.1865	0

Table-3 summarizes the controller parameters optimized using Particle Swarm Optimization for different control strategies. The optimized gains demonstrate the effectiveness of PSO in accurately capturing the dynamic behavior of the four-area power system. Compared to conventional tuning, the PSO-tuned PID controller provides improved distribution gain, resulting in better damping and faster response. Among all methods, the PSO-tuned fractional-order PID controller shows the best performance due to the additional flexibility introduced by fractional integration and differentiation orders. These extra degrees of freedom enable finer control of system dynamics, leading to reduced overshoots and shorter settling time, thereby making the PSO-based FOPID controller the most effective approach.

6 Conclusion

A systematic approach for tuning both conventional PID and fractional-order PID controllers is proposed in this work. The tuning strategy combines classical and modern techniques, where the Ziegler–Nichol's method is employed to determine the proportional and integral gains, while the Astrom–Hagglund method is used for estimating the derivative gain of the conventional PID controller. Furthermore, the Particle Swarm Optimization (PSO) technique is applied to optimally tune the parameters of both PID and fractional-order PID controllers with the objective of improving the closed-loop system response.

Simulation results clearly demonstrate that the fractional-order PID controller tuned using PSO (PSO-FOPID) exhibits superior performance compared to the PSO-tuned PID controller and the conventional PID controller. The proposed fractional-order controller shows improved transient characteristics, enhanced stability, and better robustness. In particular, the fractional-order PID controller provides a stable operating region even in cases where the integer-order PID controller fails to ensure stability.

The effectiveness of the proposed approach is further validated by the parameter optimization capability of the modified PSO algorithm. The optimization process efficiently searches for the global optimal solution within a predefined parameter space, ensuring improved controller performance. These results indicate that PSO-based fractional-order controller design is a promising approach and can be effectively extended to various control applications in future research.

References

[1] J.G. Zeigler and N.B. Nichols," Optimum setting for automatic controller', ASME Trans., pp. 759-768, 1942
[2] S. Manabe, "The non – integer integral and its application to control systems," ETJ of Japan, vol.6, no.3-4, pp.83-87, 1961.4.
[3] K.J. Astrom, CC. Hang, P. Pearson and W.K Ho:" A self-tuning controller with PID structure." Inst.J. of Control, 38, pp. 401-417(1983)
[4] K.J. Astrom and T. Haggland;" Automatic tuning of PID controllers", Instrument Society of America (1988).
[5] G. Maione and P. Lino, "New Tuning Rules for Fractional PIλDμ controllers", Nonlinear Dynamics, 2006.
[6] M. Zhuang and D P Atherton, "Automatic tunning of Optimum PID controllers," in IEEE Proceeding- Control theory and Application, vol. 140, 1993, pp.216-224
[7] K.J. Astrom and T. Haggland, "PID Controllers: Theory, Design and Tuning." Research Triangle Park, NC: International Society for Measurement and Controls, 1995
[8] Smerpitak, K., Ukakimaparn, P., Trisuwannawat, T., and Laorsri P. (2012). Discrete-time PIDA controller designed by kitti's method with bilinear transform. In: International Conference on Control, Automation and Systems (ICCAS), 2012 12th; pp.1585-1590.
[9] Zhao, J., Li, T. and Qian, J. (2005). Application of particle swarm optimization algorithm on robust PID controller tuning. Advances in Natural Computation: Book Chapter. Springer Berlin / Heidelberg, pp. 948-957Astrom, K. J. and T., Hagglund, PID Controllers: Theory, Design and Tuning, ISA, Research Triangle, Par, NC, (1995).
[10] Shi, Y.H. and Eberhart, R.C. (2001). Fuzzy Adaptive Particle Swarm Optimization. Proc. Congress on Evolutionary Computation. Seoul, Korea.
[11] Kao, Chih-Cheng, Chin-Wen Chuang and Rong-Fong Fung 2006 The self-tuning PID control in a slider – crank mechanism system by applying particle swarm optimization approach Mechatronics 16.8 513-522.
[12] Q.Wang, P Spronck and R Tracht. (2003). An Overview of Genetic Algorithms Applied to Control Engineering Problems. Proceedings of the Second International Conference on Machine Learning and Cybernetics.
[13] Gaing, Z.L. (2004). A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Transaction on Energy Conversion, Vol.19(2), pp.384-391.