Int. j. inf. tecnol. (October 2022) 14(6):2877–2884
https://doi.org/10.1007/s41870-022-01051-3

ORIGINAL RESEARCH

Design of PSO tuned PID controller for different types of plants

**Khamma Kanwar[1] · Jayashri Vajpai[1] ·
Santosh Kumari Meena[1]**

Received: 20 April 2022 / Accepted: 28 July 2022 / Published online: 25 August 2022

Abstract Several innovative techniques have been developed recently for the intelligent design of the widely used PID controllers for industrial systems where accuracy is of paramount concern. In addition to the classical methods, these include soft computing based methods such as Fuzzy logic, Neural Network, Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). PSO is a well-established computational technique based on swarm intelligence. This method is motivated by the observation of social interaction and behavior of animals with limited brains in solving complex problems such as fish schooling and bird flocking. In this paper, PID controller parameters have been designed by using PSO for three different types of plants represented by their transfer functions and compared with those obtained by using the standard Ziegler–Nichols (Z–N) method. The proposed methodology was implemented using MATLAB/Simulink software, and the step response characteristics obtained by the application of Z-N method based PID controller and PSO based PID controller have been analyzed. The performance of both strategies is compared in terms of PID controller tuning parameters, peak overshoot (%), and settling time. It has been found that PSO based controller performs better than Z-N based controller for all three plants.

Keywords Particle swarm optimization (PSO) · Swarm computing · Ziegler–Nichols (Z–N) method · PID controller · PSO tuned PID controller

1 Introduction

Industries like electrical, mechanical, construction, petroleum, iron, steel, cement, power and paper need higher productivity, thus demanding better automation of production line mechanisms driven by electrical motors. They suffer from problems in operation due to instability whenever there is a need for sudden changes in position or speed dynamics.

To overcome these problems, an intelligent control system is the needed for operation of industrial machines so that the desired control targets could be achieved. Proportional Integral Derivative (PID) controllers are hence widely used in industries for process control applications. The merit of using PID controllers lies in their simplicity of design and good performance, including low percentage overshoot and small settling time making them suitable for slow industrial processes [1]. Despite the PID controller's simple construction and resilience, effective tuning of gains K_p, K_I, and K_d is often problematic. The classical methods are useful for obtaining the starting values for tuning the PID parameters but are often limited in their performance since accurate plant parameters are seldom available. Therefore, the benefit of using a modern optimization approach is harnessed as a complementary solution to fine tune the parameters of the PID controller designed by the conventional method. There are many emerging techniques that can be used for this purpose. These include soft computing and swarm computing methods. PSO is one of the latest and among the most efficient optimization tools in the category of swarm intelligence.

✉ Khamma Kanwar
kkjoya.jnvu@gmail.com

Jayashri Vajpai
jayashrivajpai@gmail.com

Santosh Kumari Meena
sanumeena11245@gmail.com

[1] Department of Electrical Engineering, M. B. M. University, Jodhpur, Rajasthan 342011, India

Swarm intelligence (SI) is a branch of multidisciplinary artificial intelligence that studies collective behavior and emergent properties of complex, self-organized, decentralized systems with social structure. Examples of such systems are ant colonies, bird flocking, animal herding, bacterial growth and fish schooling. It can deal with intricacies of control of nonlinear systems that cannot be accurately described mathematically. It has overlapping applications with soft computing and recently evolved into a multidisciplinary system that combines Fuzzy Logic, Neural Networks, Genetic Algorithm (GA), Evolutionary programming (EP), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) [2]. These techniques have proven to be valuable in controller design for complex model-defying plants, as revealed by the study of literature.

J. Kennedy and R. Eberhart developed PSO in 1995, drawing inspiration from both the adjacent fields of evolutionary algorithms and artificial life techniques. This method has many advantages, including ease of implementation, consistent convergence characteristics, and high computing efficiency [3]. It is based on concepts inspired by the activity of birds, insects, and their societies in miniature, focusing on the fact that they work as a team rather than individually. PSO combines socio-cognition with evolutionary computation according to E. Ozcan and Mohan [4]. The particles are randomly placed in the search space and move with a stochastically modified velocity iteratively. Each particle has a recollection of the best location it has ever encountered, which it shares with a group of particles known as its neighborhood. A new parameter called inertia weight, was introduced by Y. Shi and R. Eberhart, into the original PSO algorithm in order to improve its performance [5].

The comparison of PSO with GA for an economic multi-objective optimization problem has been attempted by Lei et al. [6]. They have applied two upgraded intelligent optimization approaches one is multi-objective RNPGA (Random Niche Pareto Genetic Algorithms), and the other is PSO improved with linear inertia weight reduction. The PSO method solves the problem faster, as evidenced by the results.

Vaishnav and Khan have designed PID controllers for a second order system with monotonic step response using six different PID tuning formulas. They have found that a hybrid fuzzy PD plus conventional I (FPD + I) controller provides better performance among all these methods [7].

Solihin et al. [8] have designed a PID controller for a third order system by using PSO which is optimized through different performance indices. This controller produces a response with lower overshoot and better performance as compared to the classical Ziegler–Nichols (Z–N) method.

S. E. Selvan et al. have implemented the modified PSO algorithm for tuning of PID controller for different plant transfer functions and shown that settling time values for the PID controller coefficients selected by PSO was the best among all the tuning methods [9].

The GA, EP, and PSO techniques have been used by B. Nagaraj et al. to improve the performance of the PID controller for armature controlled DC motor. This tuning algorithm reduced the overshoot and rise time as compared to other PID controller tuning algorithms, such as the Z–N tuning method and continuous cycling method [10].

In another attempt, B. Nagaraj and P. Vijayakumar proposed a hybrid algorithm to design a PID controller with ACO and compared it with the conventional Z–N and Cohen Coon methods. This controller improves the performance of process in terms of time domain specifications in comparison with the conventional methods [11].

A novel PSO-based optimally designed PID controller for a linear brushless DC motor was proposed by M. Nasri et al. [12]. The proposed method outperformed the GA and Linear quadratic regulator (LQR) in terms of improving step response characteristics such as steady-state error, rise time, settling time, and maximum overshoot in speed control of a linear brushless DC motor.

Inspired by this, an effort has been made by the authors of this paper to design PSO tuned PID controllers for various plants represented by transfer functions of 2nd order, 3rd order and 4th order systems and implement them in MATLAB/Simulink software. The effect of tuning of PID controller by using the traditional Z-N method and the bio-inspired PSO technique has been compared in this study. The PSO optimized tuning parameters of the PID controller improve the time domain performance in terms of settling time and peak overshoot as compared with the Z-N method. Further, it is noted that the PID controller tuning for such systems by using traditional techniques is time-consuming and often leads to significant errors in the performance characteristics.

2 Design of PID controller

The PID controllers are widely used for industrial applications, as they provide optimal and robust performance not only for stable systems but also for unstable and nonlinear systems. The controller maintains the output in the desired or required range to get the best possible performance by tuning its parameter values. If the plant operating characteristics are dynamic in nature, the PID parameters have to be tuned frequently. With parametric change, the controlled system may perform poorly or may become unstable due to improper values of the controller parameters. Hence, it is mandatory to tune the controller parameters for dynamic systems from time to time to achieve good performance. This can be done automatically through algorithms.

The tunable PID controller continuously adjusts the control inputs in an attempt to reduce the error i.e. the difference

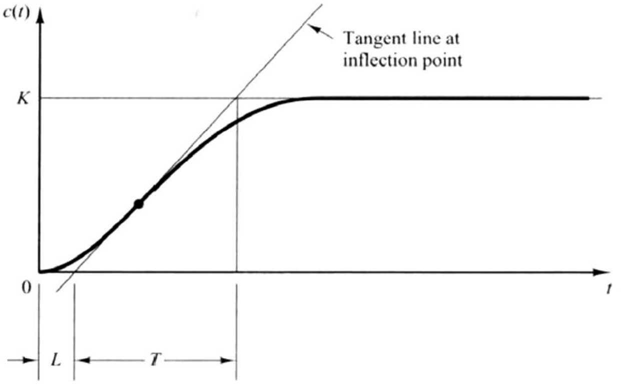

Fig. 1 S-shaped response of unit step

Table 1 Tuning rule based on step response of plant [13]

Type of Controller	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	2L	0.5L

between a measured process output variable and the desired set point.

The transfer function G_c (s) of the PID controller can be described as:

$$G_c(s) = K_p + \frac{K_i}{S} + K_d s \quad (1)$$

By tuning the three parameters K_p, K_i and K_d i.e. proportional, integral and derivative gain values in the PID controller, it can provide the necessary control action for specific process requirements.

2.1 Ziegler-Nichols method

The well established Z-N method can be employed for the design of PID controllers used for both closed and open loop systems, as described in the following sub sections.

2.1.1 *Open loop process reaction curve method*

The plant's experimental response to a unit step input is measured and assessed. If there is no integrator or dominant complex-conjugate pair of poles in the plant, the unit-step response curve will be shaped like a 'S' and can be described by two constants, delay time (L) and time constant (T) as shown in Fig. 1[13].

The setting of the values of K_p, T_i, and T_d is done according to the Z-N formula described in Table 1.

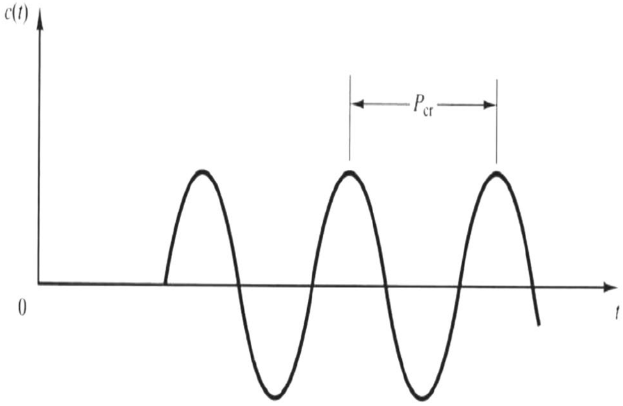

Fig. 2 Sustained oscillation with period P_{cr}

Table 2 Tuning rule based on K_{cr} and P_{cr} [13]

Type of controller	K_p	T_i	T_d
P	0.5 K_{cr}	∞	0
PI	0.45 K_{cr}	$\frac{1}{1.2}P_{cr}$	0
PID	0.6 K_{cr}	0.5 P_{cr}	0.125 P_{cr}

2.1.2 *Closed-loop ultimate gain method*

With the values of $T_i = \infty$ and $T_d = 0$. This approach obtains the critical gain K_{cr} at which the first sustained oscillations are achieved in the step output by increasing the proportional control action K_p and corresponding period $P_{cr,}$ is experimentally determined from Fig. 2. This method is not applicable if the output does not exhibit sustained oscillations for any value of K_p [13] (Figs. 3, 4). The Z-N formula gives the values of parameters K_p, T_i and T_d as depicted in Table 2.

The formulae given in Tables 1 and 2 have been derived empirically for both the methods suggested by Z-N. They have proven to be quite successful in designing a controller that produces accurate responses for low order, linear, stable systems, but lose their relevance for higher order, nonlinear and unstable systems. Such systems require alternative methods for the design of controllers. These include intelligent soft computing and swarm intelligence techniques, such as PSO, which have been discussed in depth in the next section of this paper.

3 Particle swarm optimization (PSO)

PSO is biologically inspired swarm intelligence based computing approach that follows the tenets of social interaction behavior of fish schooling and bird flocking. Its success in engineering applications is primarily due to its high computational efficiency, easy implementation and stable convergence. Another advantage of the PSO is that it does not need a detailed mathematical description of the process and along with the controller tuning, it also supports the retuning of controller parameters, if necessary [14].

In this approach, the search space population is known as a swarm and every solution is represented as a "Particle". All particles have fitness values which are obtained by computing the fitness function at the current positions, as well as velocities that guide the particles to flight. The particles keep flying through the problem space by following the particles with the best solutions so far [15]. Thus each possible solution can be modeled in the real number space as a location of a particle that moves through the problem hyperspace. The position of each particle is determined by the vector X_i Ɛ R^n and its movement by the velocity of the particle V_i Ɛ R^n as shown in Eq. (2) & Eq. (3).

$$X_{id}(\mathrm{n}+1) = X_{id}(\mathrm{n}) + V_{id}(\mathrm{n}+1) \tag{2}$$

Each particle's position and velocity are updated based on the personal past experience of the decisions made so far and the success of these decisions, as well as their knowledge of the performance of others in the neighbourhood. Because the relative importance of these two criteria varies from one decision to the next, it is appropriate to assign random weights to each, therefore the velocity of the $\mathrm{i.}^{th}$ particle in d dimension space with (n + 1) iteration will be decided by [16]

$$V_{id}(n+1) = w * V_{id}(n) + c_1 * rand_1() * \left(pbest_{id}(n) - X_{id}(\mathrm{n})\right) + c_2 * rand_2() * \left(gbest_{gd}(n) - X_{id}(n)\right) \tag{3}$$

where, w is the inertia weight factor., c_1 and c_2 are positive numbers, called Acceleration constant. $rand_1()$ and $rand_2()$ are two random numbers independently generated. $pbest_i$ is the best position that the corresponding particle has found so far. $gbest_g$ is the best position of the entire swarm.

The general PSO algorithm flow chart is depicted as follows:

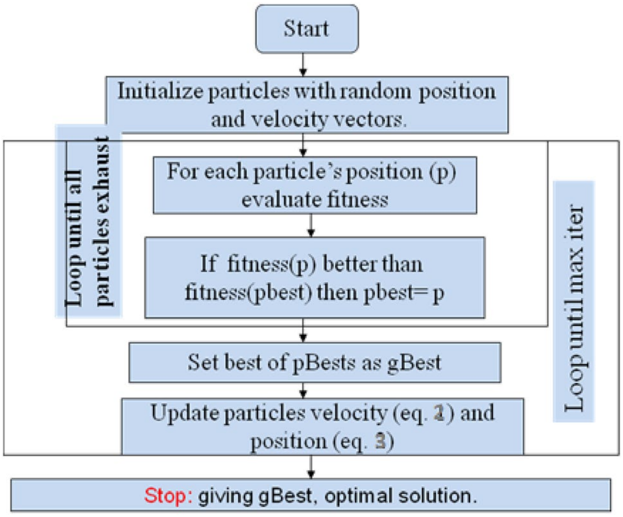
Start
Initialize particles with random position and velocity vectors.
For each particle's position (p) evaluate fitness
If fitness(p) better than fitness(pbest) then pbest= p
Loop until all particles exhaust
Set best of pBests as gBest
Update particles velocity (eq. 2) and position (eq. 3)
Loop until max iter
Stop: giving gBest, optimal solution.

Fig. 3 Flowchart of PSO algorithm

The working of PSO depends upon proper determination of fitness function for selecting the best performance. This varies from application to application. The design of the fitness function for this work is explained after the problem formulation.

4 Problem formulation and methodology

The Type and Order of higher order plants further increase with the application of PID controller. The determination of controller parameters for such plants is a challenging task. The traditional Z-N method requires accurate values of plant parameters, which are seldom available for complex plants. Hence, the design of controller using this approach does not produce desirable output at times. In such cases, PSO tuned PID controller algorithm has been developed to improve the transient response of the system.

The tuning of the PID controller by Z-N method and PSO algorithm is carried out in this work. The aim is to determine the three optimal values of the controller parameters K_p, K_i and K_d for three different types of plants represented by their transfer function. The objective of the PSO based design of PID controller is to seek a set of PID parameters such that the feedback control system improves time domain specification with minimum overshoot. The effectiveness of the PSO based PID controller design depends on the construction of an efficient fitness function. Hence, the fitness function has been elaborated separately in the sub Sect. 4.2.

4.1 Development of plant and controller models in MATLAB

The experimental set up includes the closed loop PID controller cascaded with the plant to be controlled. The controller was tuned for the optimal values of K_p, K_i, and K_d first by using the Z-N method and then by the PSO algorithm. A MATLAB/Simulink model has been developed for the simulation as shown in Fig. 4.

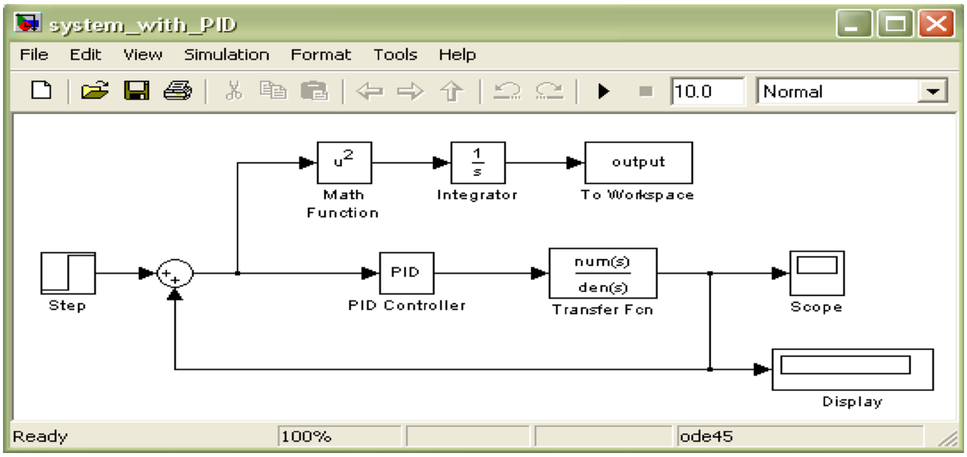

Fig. 4 Simulink Model for tuning of PID controller using ISE

The same simulation model will be used for the implementation of other plants where only transfer function block will change. Similarly, the controller parameters are changed for the simulations: first according to the Z–N approach explained in the Sect. 2 and then according to the PSO approach explained in Sect. 3.

4.2 Design of fitness function for PSO

The optimal values of the control parameters K_p, K_i and K_d, are selected to achieve a good step response along with minimization of performance criteria in the time domain according to defined fitness function. The system step response is utilized to determine the error. Iterations are carried out till the error becomes as small as desired. The fitness function can be described in terms of peak overshoot, steady state error, settling time, and rise time as shown in Eq. 4

$$F = (1 - exp(-0.5)) * (M_p + E_{ss}) + exp(-0.5) * (t_s - t_r) \tag{4}$$

where M_P = peak overshoot. E_{ss} = steady state error. t_s = settling time. t_r = rise time.

4.3 Implementation of PSO tuned PID controller

The PSO technique is used to find out the optimal parameters of PID controller for three different types of plants represented by their transfer function. The parameters used to design the PSO and tune the controller parameters for all these plants are shown in Table 3.

The PSO algorithm is run with these parameters for the maximum iteration count of 20, to determine the optimal values of PID gains for the selected 2nd order, 3rd order and 4th order plants as described in the next section.

5 Simulation results for different types of plants

This section describes the comparative analysis of the performance of the selected systems of 2nd, 3rd and 4th order by employing PID controller whose values of the gains were derived by first using the Z-N method and then by using the PSO algorithm. These plants have been selected from the papers by Vaishnav and Khan [7], Solihin et al. [8], and Selvan et al. [9] respectively, so that validation of results can be done properly. These plants are expressed in Eqs. (5), (6) and (7) as follows:

$$G(s) = \frac{1}{s^2 + 10s + 20} \tag{5}$$

$$G(s) = \frac{1}{s^3 + 9s^2 + 23s + 15} \tag{6}$$

$$G(s) = \frac{5}{s^4 + 3s^3 + 7s^2 + 5s} \tag{7}$$

Simulation of these plants are carried out in MATLAB software and step response characteristics obtained by the application of Z–N based PID controller and PSO tuned PID controller have been shown in the following plots respectively (Figs. 5, 6, 7, 8, 9, 10).

Table 3 Parameter for PSO [3]

Parameter	Values
No. of particles	20
Maximum no. of iterations	20
Inertia weight factor (w)	0.6
Acceleration constant (C_1)	1.5
Acceleration constant (C_2)	1.5

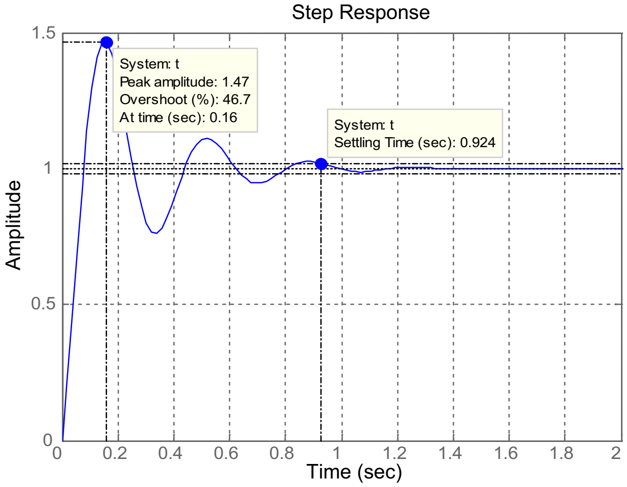

Fig. 5 Step response of a 2nd order system with PID controller tuned by Z–N method

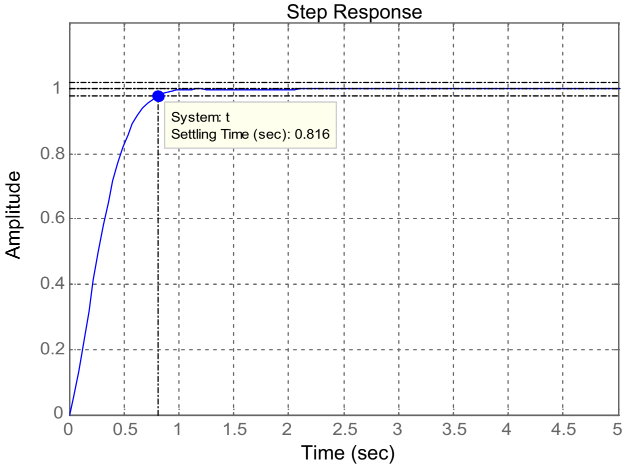

Fig. 6 Step response of the 2nd order system with PID controller tuned by PSO method

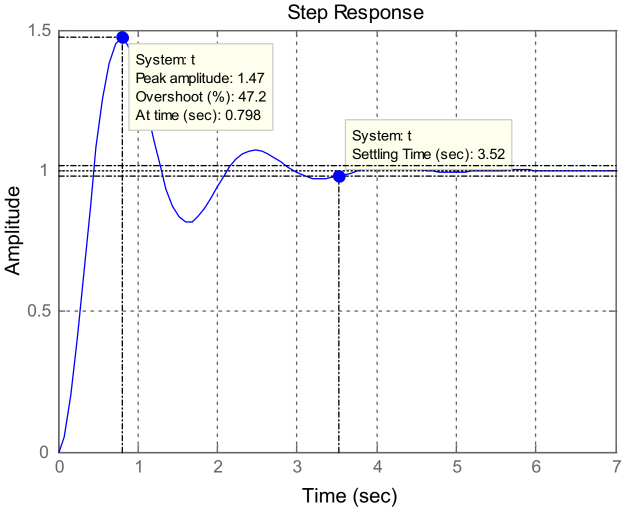

Fig. 7 Step response of a 3rd order system with PID controller tuned by Z–N method

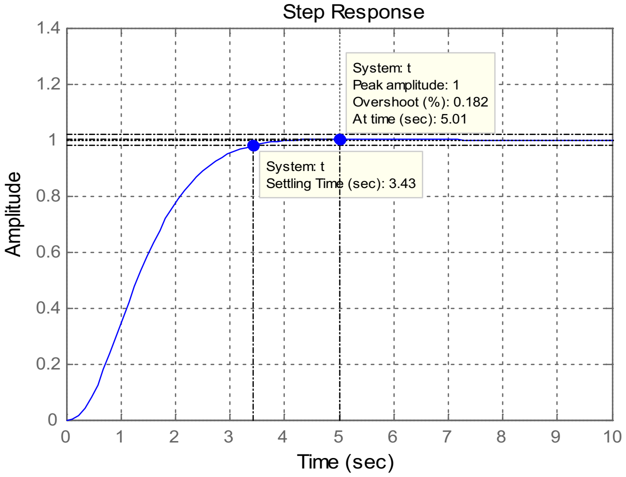

Fig. 8 Step response of the 3rd order system with PID controller tuned by PSO method

The PID controllers for each of the three plants as described in the previous section were tuned by Z–N method in the first part of this work. It can be seen that despite keeping the same parameters there is some deviation in results obtained by simulation from those obtained by the respective authors [7–9]. The step responses of 2nd, 3rd and 4th order systems with PID controller tuned by Z–N method are shown in Figs. 5, 7 and 9 respectively.

Then PSO algorithm has been employed for tuning K_p, K_i, and K_d parameters of PID controller. It optimizes the value of a fitness function with consideration of the optimal peak overshoot and settling time of the plants represented by the transfer function. The results for all these simulations have been compared as depicted in Table 4. The step responses of these systems with PSO tuned PID controller are shown in Figs. 6, 8 and 10 respectively.

It can be seen that by employing PSO based PID controller in 2nd order, 3rd order and 4th order systems, the plant step response produces lesser peak overshoot and takes less settling time than the controller designed by using the classical Z-N method by the respective authors. It is observed that there are some variations in the simulation results obtained in this work from those reported by the authors despite the fact that the same values of controller parameters were used. The values of PSO tuned PID controller parameters K_p, K_i, and K_d are also lesser than those obtained by Z–N based PID controller method.

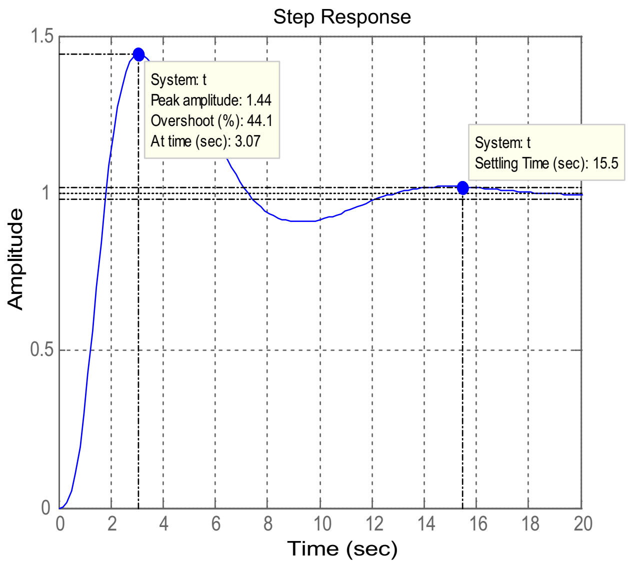

Fig. 9 Step response of a 4th order system with PID controller tuned by Z–N method

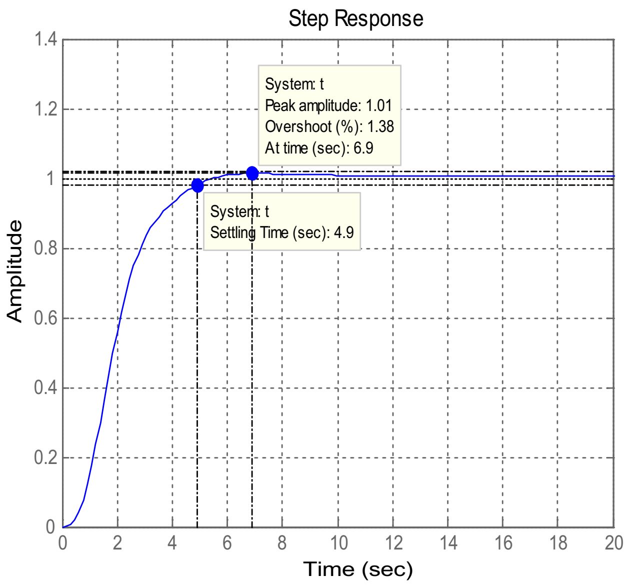

Fig. 10 Step response of the 4th order system with PID controller tuned by PSO method

6 Conclusion

There are many emerging techniques that can be used for tuning of PID controller parameters for industrial applications. The classical methods are good for defining the starting point for tuning of the PID controller values, but these tuning methods required an accurate plant model, which is seldom available. The PSO is one of the most efficient optimization nature-inspired computing tools. The design of the PID controller for three different types of plants represented by their transfer functions, has been implemented in MATLAB/Simulink software by first using the Z-N approach and subsequently the PSO algorithm to tune the closed loop PID controller cascaded with the plants for the values of K_p, K_i, and K_d. In both these situations, the peak overshoot and settling time have been calculated. As indicated by the comparison of results in Table 4, the tuning parameters, peak overshoot and settling time achieved by the PSO tuned PID controller are substantially smaller than the values acquired by the Z-N based PID controller in all circumstances. On the basis of simulation results for a variety of plants, it may be concluded that PSO tuned PID controller gives better

Table 4 Comparison of results obtained by Z–N method and PSO algorithm

Method	2nd order system $\frac{1}{s^2+10s+20}$		3rd order system $\frac{1}{s^3+9s^2+23s+15}$		4th order system $\frac{5}{s^4+3s^3+7s^2+5s}$	
	Results reported by Vaishnav and Khan [7]	Simulation results	Results reported by Solihin et al. [8]	Simulation results	Results reported by Selvan et al. [9]	Simulation results
Z–N Method						
K_p	387	387	115.2	115.2	0.9042	0.9042
K_I	3722	3722	175.9	175.9	0.3603	0.3603
K_D	10	10	18.9	18.9	0.6939	0.6939
Peak overshoot (%)	58.5	46.70	67	47.2	–	44.1
Settling time (s)	0.77	0.924	4.2	3.52	11.7143	15.5
PSO algorithm						
K_p	–	27.68	95.6	10.8231	0.2453	0.4905
K_I	–	65.11	14.6	10.4638	0.0016	0.0015
K_D	–	0.90	10.9	0.1773	0.0383	0.2982
Peak overshoot (%)	–	0.00	3.8	0.18	–	1.38
Settling time (s)	–	0.816	2.6	3.43	2.33	4.9

transient response characteristics than the classical Z-N tuning method.

6.1 Scope of future work

The design of controller is an important task for industrial control systems. The classical techniques for designing of controller require accurate values of parameters of the system to be controlled. However, the swarm intelligence and the soft computing techniques do not require the mathematical model of a physical system. Hence, these techniques can be applied for tuning of controller parameters for such systems.

In this paper general PSO algorithm has been used but there are various variants of PSO algorithm that can be used for real physical systems. Further, it would be interesting to apply an extended version of PSO with evolutionary algorithms such as GA, Memetic algorithms, ACO and shuffled frog leaping.

References

1. Deepyaman M, Sagnik B, Amit K (2008) Design of a fractional order PID controller using particle swarm optimization technique. 2nd National Conference on Recent Trends in Information Systems (ReTIS)
2. Umarani R (2010) Particle swarm optimization evolution, overview and applications.Int J Eng Sci Technology Vol. 2(7), PP. 2802–2806, 2010. ISSN: 0975–5462.
3. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, (Perth, Australia), Vol. IV, pp. 1942–1948. DOI: 0–7803–2768–3/95.
4. Ozcan E, Mohan CK (1998) Analysis of a simple particle swarm optimization system. Research Gate
5. Y. Shi and R. Eberhart, “A Modified Particle Swarm Optimizer”, IEEE, pp. 69–73, 1998. DOI: 0–7803–4869–9/98.
6. X. Lei, Z. Shi, and L. Wang, “Optimization and Simulation of a MO Problem Solved by GAs and PSO”, 3rd International Conference on Natural Computation (ICNC), IEEE, 2007. DOI: 0–7695–2875–9/07.
7. S. R. Vaishnav, and Z. J. Khan, “Performance of tuned PID controller and a new hybrid fuzzy PD + I controller”, World Journal of Modeling and Simulation, World Academic Press, World Academic Union, Vol. 6 No. 2, pp. 141–149, 2010. ISSN 1 746–7233.
8. M. I. Solihin, L. F. Tack and M. L. Kean, “Tuning of PID Controller Using Particle Swarm Optimization (PSO)”, Proceeding of the International Conference on Advanced Science, Engineering and Information Technology, pp 458–461, 2011. ISBN 978–983–42366–4–9
9. S. E. Selvan, S. Subramanian, and S. T. Solomon “Novel Technique for PID Tuning by Particle Swarm Optimization”. https://www3.nd.edu/~swarm03/Program/Abstracts/SelvanSwarm2003.pdf
10. B. Nagaraj, S. Subha, and B. Rampriya, “Tuning Algorithms for PID Controller Using Soft Computing Techniques”, IJCSNS International Journal of Computer Science and Network Security, Vol. 8 No. 4, 2008.
11. Nagaraj B, Vijayakumar P (2011) A comparative study of PID controller tuning using GA, EP, PSO and ACO. J Autom Mob Robot Intell Syst 5(2)
12. Nasri M, Nezamabadi-pour H, Maghfoori M (2007) A PSO-based optimum design of PID controller for a linear brushless DC motor. World Academy of Science, Engineering and Technology
13. Ogata K (2002) Modern control engineering. Pearson Education Publisher, 4th edition
14. Rajinikanth V, Latha K (2012) Tuning and retuning of PID controller for unstable systems using evolutionary algorithm. In: International scholarly research network isrn chemical engineering, pp 1–10, 2012. DOI: https://doi.org/10.5402/2012/693545.
15. Hu X, Shi Y, Eberhart R (2004) Recent advances in particle swarm. IEEE, pp 90–97. DOI: 7803–8515–2/04
16. Eberhart RC, Shi Y (2001) Particle swarm optimization, application, and resources. IEEE, pp 81–86. DOI: 7803–6657–3/01