

A PID controller parameter tuning method based on improved PSO

Shuyue Wu*

Hunan Institute of Information Technology, Changsha, China

Received: 26-September-2017; Revised: 14-December-2017; Accepted: 16-December-2017
©2018 ACCENTS

Abstract

Proportional integral derivative (PID) controllers have been used for industrial processes for long, and PID tuning has been a field of active research for a long time. An interactive, evolution, particle swarm optimization (IEPSO) algorithm was proposed based on linear weight decrease particle swarm optimization (LWDPSO) and stochastic particle swarm optimization (SPSO). The particle swarm was divided into two groups that is standard PSO and SPSO employed for global search and local search respectively. Parallel variables were dynamically adapted according to the evolution stage. The simulations proved that the IEPSO had better performance than LWDPSO and SPSO-PID controller tuning test proved IEPSO had the better control effect than Ziegler-Nichols, LWDPSO and SPSO.

Keywords

IEPSO, PSO, PID, Fitness, Tuning, Punitive measures.

1.Introduction

Proportional integral and Derivative (PID) controllers have been used in industrial control applications for a long time. PID controllers date to 1890s governor design [1, 2]. Despite having been around for a long time, the majority of industrial applications still uses PID controllers. According to a survey in 1989, 90% of process industries use them [3]. This widespread use of PID in industry can be attributed to their simplicity and ease of re-tuning on-line [4].

In industrial control, the most widely used controller is PID controller; it has a simple structure, robustness and wide applicability. Self-tuning PID controller and the optimization of parameters has become an important research topic [5-12].

There is Ziegler-Nichols in the classic PID parameter tuning method [13]; the control tuning effect is often difficult to meet the control requirements. In [14] a method with the PID controller parameters decided based on a FOLPD model has been designed.

The main design requirement is the rejection of load disturbances. Artificial intelligence methods have been introduced, particle swarm optimization (PSO) algorithm is proposed for PID tuning parameters [15, 16].

Artificial immune systems (AIS) are computational systems inspired by the principles and processes of the vertebrate immune system, which learns about the foreign substances to defend the body against them [17]. Kim and Cho used the algorithm for disturbance function based tuning [18]. The immune algorithm is also used for auto tuning of PID controllers [19]. Ant colony optimization (ACO) is a recently developed meta-heuristic approach to solving optimization problems based on working on an ant colony [20, 21]. Ant colony optimization was used for PID tuning [22]. It was used to minimize a multi-objective function and its results were found to be better than genetic algorithm and Ziegler Nichols method. The bee's algorithm is used to tune a PID controller and solving complex systems [23]. The results of ACO, PSO and bees algorithm are compared and presented in [24]. They have suggested ACO in case of minimum overshoot and PSO in the case of fast rise time and settling time points.

In PSO algorithm, an important parameter is the inertia weight (w), if w is large, the algorithm, global search capability is strong, when w is small, the algorithm's local search ability is strong. In the early, evolution should be based on global search; evolution is mainly based on local search in the late. Linear weights dropped PSO (LWDPSO) sets w in decreases linearly strategy with the evolution conduct [25], the algorithm improves search efficiency, but it has a drawback of a premature convergence, and it is

*Author for correspondence

trapped in a local optimum. Stochastic PSO algorithm (SPSO) is with a partial search capability [26]. It results in a random location and breakthroughs search blind spots, but there are unstable characteristics. An improved PSO algorithm is presented in this paper; it is an interactive evolutionary PSO algorithm (IEPSO). In this algorithm, the swarms will be divided into two groups [27, 28] for joint search, two swarm particles were standard PSO and SPSO, they are parallel interaction evolution. Where larger w is set for the standard PSO particle group. It is focusing on global search, the SPSO particle group focuses to find the global optimum location in the vicinity of the standard PSO particle group. It works for fine search, and generates a random location, and breakthroughs search blind spots. Particles are dynamically allocated to two groups according to the stage of evolution. According to an evolutionary generation number, the number of the evolution particles is adjusted dynamically in two algorithms, to ensure that it is mainly a global search in the initial algorithm; it is based on local search later.

2. Materials and methods

2.1 Standard PSO algorithm

PSO algorithm originated from birds foraging behavior simulation, optimal population is made through collective collaboration between individuals. Standard PSO algorithm initialization generates a group of particles; each particle flight to a certain speed in n -dimensional space, flight speed is dynamically adjusted by individual flight experience and flight experience of groups. $X_i = (X_{i1}, X_{i2}, \dots, X_{in})$ is the current position of i -th particle, $V_i = (V_{i1}, V_{i2}, \dots, V_{in})$ is the current velocity of i -th particle, $P_i = (P_{i1}, P_{i2}, \dots, P_{in})$ is the ever experienced best location of i -th particle, i -th particle has the best fitness in the position. $f(x)$ is set as the minimized objective function, the best position of i -th particle is decided by the formula (1):

$$P_i = \begin{cases} P_i & f(P_i) < f(X_i(t+1)) \\ X_i(t+1) & f(P_i) \geq f(X_i(t+1)) \end{cases} \quad (1)$$

The best position in all particles position is called global best position, it is denoted as $P_g(t)$. That is formula (2).

$$P_g(t) = \arg \min \{ f(P_i) \mid i = 1, 2, \dots, S \} \quad (2)$$

PSO algorithm evolution equation is in (3), (4) formulas.

$$V_i(t+1) = wV_i(t) + c_1r_1(P_i - X_i(t)) + c_2r_2(P_g - X_i(t)) \quad (3)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (4)$$

Where $i = (1, 2, \dots, S)$, S is the size of the population; c_1, c_2 is accelerating factor; $r_1, r_2 \in [0, 1]$ is a random value of uniformly distributed; w is the inertia weight, if w is large, global search ability is strong, when w is small, the local search ability is more prominent. LWDPSO set w with the linear decrease evolution, w is in the range of $[0.4, 0.95]$, the relationship between w and the number k of evolution can be defined as (5) formula.

$$w = 0.95 - \frac{k}{k_{\max}} \times 0.55 \quad (5)$$

Where w is the maximum number of evolution. After the end of the evolution, the optimal location in the PSO algorithm search is the optimal solution of the system requirements.

2.2 SPSO algorithm

SPSO algorithm is the particle velocity evolution equation (3) in order inertia weight $w = 0$, namely:

$$V_i(t+1) = c_1r_1(P_i - X_i(t)) + c_2r_2(P_g - X_i(t)) \quad (6)$$

In this case, the algorithm, global search is weakening, and local search capability is enhancement. In the current global optimum position, j -th particle velocity will be zero, it stops the search, in order to enhance the global search ability, the historical best position P_g is preserved in PSO, and in the search space, the re-location of randomly generated j -th particle is P'_j , the self-position of j -th particle is preferably updated to the formula (7):

$$P_j = \begin{cases} P_g & f(P_g) < f(P'_j) \\ P'_j & f(P_g) \geq f(P'_j) \end{cases} \quad (7)$$

In this evolution, the other particles evolve with the velocity and position in accordance with formula (4), (6), and by the formula (1), their best place is obtained by the update after the compared their results. After each iteration, the particles at the global optimal position generate randomly a new position again, to repeat the operation until the end of evolution.

SPSO has local search ability, its global optimization ability relies on random position which the current particles of the best location generated, they break the blind search algorithm, but it lacks of stability.

2.3IEPSO algorithm

An improved PSO algorithm is presented in this paper that is PSO interactive evolutionary algorithm (IEPSO). The number of particles is S , the total evolutionary generation is G , the current evolutionary generation is i , particle swarm is divided into two groups, evolution is made respectively by using standard PSO and SPSO. The standard PSO particle group focuses on global search, SPSO particle group focused on fine search in the area where the standard PSO particle group finds the global optimum location, and it generates a random location and breakthroughs blind spots. When a standard PSO evolution is used, the number of particles is $S * (G-i) / C$ (rounding toward minus infinity direction), when SPSO evolution is used, the number of particles is $S * i / G$ (rounded towards positive infinity direction).

When the initial evolution, the number of particles is largest in the standard PSO particle group, and it reaches to $S-1$. In global search, SPSO particle group only has one particle, when the area of the current optimal location is found in the global search, then local search is done. It is expected to search better position in the region. In evolutionary late dynamic particle is gradually adjusted from the standard PSO particle group to the SPSO particle group. The space has been multiple searched by a standard PSO particle group. The SPSO particle group makes local search for the small areas of current best positions, and it generates a random location and breakthroughs search blind points. Standard PSO particle group searches in the relevant large area of the current global optimal spatial position, it does not give up any opportunity to find the global optimum position. In order to ensure that the standard PSO algorithm

has a large global search capability, IEPSO inertia weight W value takes one.

3.Results

3.1Experiments and analysis

The four typical test functions [14] Sphere function, Rastrigrin function, Rosenbrock function and Griewank function have been used to test performances of LWDPSPSO, SPSO and IEPSO as the fitness function for testing. In order to avoid the impact of the initial population location of algorithm performance, asymmetric shape search space is selected in Sphere function and Rastrigrin function, and symmetric search space is selected in Rosenbrock function and Griewank function. Their dimension and the search ranges in the typical test functions have shown in *Table 1*.

Table 1 Typical test functions

Function	Dimension	Range
Sphere	30	[-50,100]
Rastrigrin	20	[-5.2,10.24]
Rosenbroek	30	[-30,30]
Griewank	100	[-600,600]

In each PSO algorithm, the maximum evolutionary number is 500 generations. The size of the population is 80, c_1 and c_2 is equal to 2. To avoid accidental experiment phenomenon, now three algorithms simultaneously carry out 10 experiments for these four functions. These four algorithms, optimization, performance comparison graphs with the fitness and evolution has been shown in *Table 2*. *Table 2* shows the adaptation of the average, maximum and minimum values which are obtained by the three algorithms in 10 experiments.

Table 2 Comparison of fitness optimization

Function	Algorithm	Average	Maximum value	Minimum value
Sphere	LWDPSPSO	0.0138	0.0300	0.0029
	SPSO	7.59E-06	2.49E-05	9.32E-09
	IEPSO	1.16E-06	3.70E-06	1.46E-13
Rastrigrin	LWDPSPSO	14.6048	19.813	10.3697
	SPSO	7.62E-04	0.0029	1.77E-05
	IEPSO	1.22E-05	5.73E-05	1.08E-07
Rosenbrock	LWDPSPSO	27.5945	30.0881	11.3921
	SPSO	4.68E-04	0.0016	3.41E-05
	IEPSO	5.27E-05	1.96E-04	8.50E-07
Griewank	LWDPSPSO	0.0112	0.0177	0.0071
	SPSO	1.95E-04	6.48E-04	1.23E-05
	IEPSO	1.46E-05	8.39E-05	9.87E-08

The LWDPSPSO search capabilities are poor, early signs are very clear. SPSO pre particle velocity is smaller, particles centralize in the local small-

scale. It is possible to find the optimum position in the early, middle and late optimization depends on whether to find the optimal position randomly. The

fitness curve randomness is the larger, fitness will be jumping change, and it lacks of stability. IEP SO with two interactive evolutionary algorithms has better global search capabilities and the ability of local convergence, when the fitness curve reduces in a stable premise. There will be random small decrease. At the same time, it can be drawn from *Table 1* that LWDP SO optimization is poor. The resulting fitness is much higher than the ones in the other two algorithms. The SPSO optimization result is superior to LWDP SO, and it is inferior to IEP SO. When fitness is obtained by IEP SO simulation, its average, maximum and minimum values are the lowest in the three algorithms.

3.2PID parameters self-tuning experiment Based on IEP SO

PID controller can be described as in equation (8).

$$G_c(s) = k_p + \frac{k_I}{s} + k_D s \quad (8)$$

By tuning these three parameters k_p , k_I and k_D , the system is made to meet the required performance. System design parameters are the k_p , k_I and k_D . The individual particles may be encoded in three-dimensional space, $[k_p, k_I, k_D] = [x(1), x(2), x(3)]$. The search range of the parameters and flight speed range are set to formulas (9) and (10).

$$0 < x(1) < 4k_{p1} \quad (9)$$

$$0 < x(2) < 4k_{I1}$$

$$0 < x(3) < 4k_{D1}$$

$$-2k_{p1} < v(1) < 2k_{p1} \quad (10)$$

$$-2k_{I1} < v(2) < 2k_{I1}$$

$$-2k_{D1} < v(3) < 2k_{D1}$$

The controller parameters k_{p1} , k_{I1} and k_{D1} are obtained for the introduction of Ziegler-Nichols Tuning. In order to get the right performance of transition and dynamic properties, the appropriate fitness function J must be selected. The error and the time must be controlled in the fitness function, as well as the overshoot σ punitive measures are set in the objective function.

$$J = \alpha * \int_0^{\infty} |e(t)| dt + \beta * t_s + \gamma * \sigma \quad (11)$$

Where $e(t)$ is systematic error, t_s is regulation time, α , β and γ are weights. α , β and γ set values were 0.75, 0.25 and 5 respectively. In IEP SO algorithm, PID controller parameters k_p , k_I and k_D are mapped into particles. Selection is optimized for these parameters.

4. Discussion

The high-end system is selected as a control object simulation. In the experiment, Ziegler-Nichols method, LWDP SO, SPSO and IEP SO are used respectively to tune the PID control parameters and to compare these results. The number of initial population (PSO) is 40. The number of iterations is 100. The search and flight speed ranges are determined according to formula (9) and (10). The sampling time is 20 ms. The input signal is a step signal.

Control object is shown in formula (12).

$$G(s) = \frac{3}{s^4 + 3s^3 + 6s^2 + 3s + 1} \quad (12)$$

In *Table 3*, the controller parameters are listed by four kinds of tuning method, these parameters are obtained in optimal controller for the control object. Unit step response curve of the control object has been shown in *Figure 1*.

Table 3 PID controller tuning parameters

Tuning method	Control object		
	k_p	k_I	k_D
Ziegler-Nichols	0.5743	0.2256	0.9826
LWDP SO	0.6717	0.2431	0.9910
SPSO	0.5056	0.2329	0.8273
IEP SO	0.5352	0.2246	0.9614

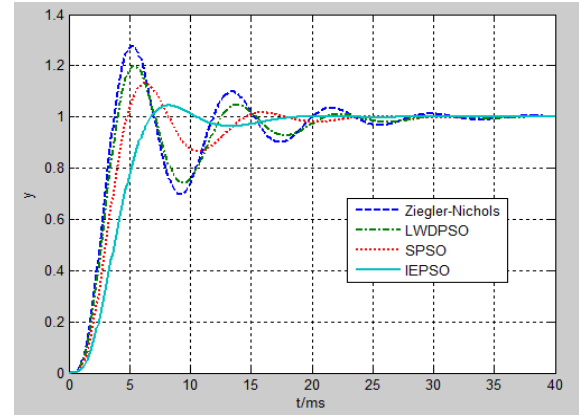


Figure 1 Step response curve comparison chart

The overshoot of the step response curves is minimal with PID controller parameters by using IEP SO optimization as shown in *Figure 1*. The overshoot of the step response curves is minimal, and its adjusting time is the shortest. While in the step response curves with LWDP SO optimization, the effect of parameters is still good such as overshoot and adjustment time. In step response curves by using SPSO optimization, SPSO overshoot is less than LWDP SO, but it is adjusted for longer than LWDP SO. In step response

curve with Ziegler-Nichols method, the overshoot is the maximum, and its adjusting time is the longest.

5. Conclusion

In this study, LWDPSSO and SPSO have been discussed. The LWDPSSO and SPSO features are analyzed. The features of the standard PSO and SPSO are researched, and the interactive evolution of PSO is proposed. The simulation results of the four fitness function have been discussed. IEPSSO optimizing performance is proven better in comparison to LWDPSSO and SPSO. IEPSSO, LWDPSSO, SPSO and Ziegler-Nichols methods are used to perform simulation experiments for the controlled object. It is proved that PID controller parameter tuning result is best in IEPSSO algorithm. PID parameter tuning has been developing continuously, random algorithm and fuzzy technology are the mainstream [29-34].

Acknowledgment

This work was supported by the Scientific Research Project (NO. 14A084) of Hunan Provincial Education Department, China.

Conflicts of interest

The author has no conflicts of interest to declare.

References

- [1] Bennett S. A history of control engineering, 1930-1955. IET; 1993.
- [2] Bennett S. Nicholas minorsky and the automatic steering of ships. IEEE Control Systems Magazine. 1984; 4(4):10-5.
- [3] Araki M. PID control: control systems, robotics and automation: system analysis and control: classical approaches II, Unbehauen, H.(Ed.). EOLSS Publishers Co. Ltd., Oxford, UK; 2009.
- [4] Åström KJ, Hägglund T. PID controllers: theory, design, and tuning. Research Triangle Park, NC: ISA; 1995.
- [5] Miller RK, Michel AN, Farrell JA. Quantizer effects on steady-state error specifications of digital feedback control systems. IEEE Transactions on Automatic Control. 1989; 34(6):651-4.
- [6] Visioli A. Tuning of PID controllers with fuzzy logic. IEE Proceedings-Control Theory and Applications. 2001; 148(1):1-8.
- [7] Seng TL, Khalid MB, Yusof R. Tuning of a neuro-fuzzy controller by genetic algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 1999; 29(2):226-36.
- [8] Mitsukura Y, Yamamoto T, Kaneda M. A design of self-tuning PID controllers using a genetic algorithm. In proceedings of the American control conference 1999 (pp. 1361-5). IEEE.
- [9] Krohling RA, Rey JP. Design of optimal disturbance rejection PID controllers using genetic algorithms. IEEE Transactions on Evolutionary Computation. 2001; 5(1):78-82.
- [10] Wang L. Application of adaptive genetic algorithms in PID controller design. Journal of Xi'an University of Science and Technology. 2005; 25(1):93-5.
- [11] Xiaowei W, Liang YX, Zhaoping L, Xiangmin J. Study on parameter tuning of fin stall PID controller based on PSO algorithm. Journal of Computer Control Technology and Applications. 2016; 23(8): 1197-202.
- [12] Heng L, Xing G, Wei L. PID controller parameter optimization based on improved glowworm swarm optimization. Computer Applications and Software. 2017; 34(7):227-30.
- [13] Ziegler JG, Nichols NB. Optimization setting for automatic controller. Transactions of the ASME. 1942; 64(11):756-69.
- [14] Cohen G, Coon G. Theoretical consideration of retarded control. Transactions of ASME. 1953; 75:827-34.
- [15] Zhi-cheng XU. Parameter tuning method of robust PID controller based on particle swarm optimization algorithm [J]. Control and Instruments in Chemical Industry. 2006; 33(5):22-5.
- [16] J Kennedy, R Eberhart. Particle swarm optimization. Proceedings of the IEEE international conference on neural networks 1995 (pp.1942-8). IEEE.
- [17] Kim DH. Tuning of PID controller using gain/phase margin and immune algorithm. In proceedings of the soft computing in industrial applications 2005 (pp. 69-74). IEEE.
- [18] Kim DH, Cho JH. Intelligent tuning of PID controller with disturbance function using immune algorithm. In annual meeting of the fuzzy information processing NAFIPS 2004 (pp. 286-91). IEEE.
- [19] Kim DH, Hong WP, Park JI. Auto-tuning of reference model based PID controller using immune algorithm. In proceedings of the congress on evolutionary computation 2002 (pp. 483-8). IEEE.
- [20] Dorigo M, Di Caro G. Ant colony optimization: a new meta-heuristic. In proceedings of the congress on evolutionary computation 1999 (pp. 1470-7). IEEE.
- [21] Dorigo M, Di Caro G, Gambardella LM. Ant algorithms for discrete optimization. Artificial Life. 1999; 5(2):137-72.
- [22] Chiha I, Borne P. Multi-Objective ant colony optimization to tuning PID controller. Proceedings of the International Journal of Engineering. 2010.
- [23] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M. The bees algorithm-a novel tool for complex optimisation. In I* PROMS virtual international conference on intelligent production machines and systems 2006 (pp.454-9).
- [24] Jones KO, Bouffet A. Comparison of bees algorithm, ant colony optimization and particle swarm optimization for PID controller tuning. In proceedings of the international conference on computer systems and technologies and workshop for PhD students in computing 2008. ACM.

- [25] Shi Y, Eberhart R. A modified particle swarm optimizer. In evolutionary computation proceedings. In international conference on World congress on computational intelligence. 1998 (pp. 69-73). IEEE.
- [26] Zeng JC, Cui ZH. A guaranteed global convergence particle swarm optimizer. Journal of Computer Research and Development. 2004; 8:1333-8.
- [27] Long W, Liang Xm, Xiao Jh, Yan G. Dynamic hierarchical hybrid particle swarm optimization algorithm [J]. Control and Decision. 2009; 10:015.
- [28] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation. 2002; 6(1):58-73.
- [29] Tiankun W. Fuzzy Self-tuning PID control of main steam temperature in thermal power boilers based on UMDA. Journal of Chinese Society of Power Engineering. 2017; 37(7): 546-51.
- [30] Shaowei F, Yunsheng Y, Jing Z. Research on control strategy of giant magnetostrictive actuator based on genetic algorithm and PID tuning. Journal of Naval University of Engineering. 2017; 29(3): 22-5.
- [31] Kai X. Modular mobile robot PID tuning feedback self-tuning control. Machinery Design & Manufacture. 2017:230-3.
- [32] Lei L, Guobao Z, Yongming H. PID parameter tuning based on bat algorithm. Control Engineering of China. 2017; 24(3):548-53.
- [33] Ling M, Jie L. PID parameters optimization with improved cooperative coevolution algorithm. Computer Technology and Development. 2017; 27(8): 37-42.
- [34] Sen R, Fei JX, Wei L. Three phase dynamic voltage restorer based on RBF_PID control. Journal of Electrical Engineering. 2017; 12(8):28-33.



Shuyue Wu is a Professor engaged in the field of electronic communication science research. He works in School of Information Science & Engineering, Hunan International Economics University, Changsha, China. His area of research and interests lies in the field of PID Controllers, Evolutionary Computations and Optimization of PID parameters.

Email:matlab_wjf@126.com