

Research Article

Advanced Fireworks Algorithm and Its Application Research in PID Parameters Tuning

Jun-jie Xue,¹ Ying Wang,¹ Hao Li,^{1,2} Xiang-fei Meng,¹ and Ji-yang Xiao¹

¹*Air Force Engineering University, Xi'an 710051, China*

²*Department of Intelligence, Air Force Early-Warning Academy, Wuhan 430019, China*

Correspondence should be addressed to Jun-jie Xue; poot-cupic-xue@163.com

Received 13 October 2015; Revised 17 May 2016; Accepted 26 May 2016

Academic Editor: Mohammad D. Aliyu

Copyright © 2016 Jun-jie Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Proportional-Integral-Derivative (PID) controller is one of the most widely used controllers for its property of simplicity and practicability. In order to design high-quality performances PID controllers, an Advanced Fireworks (AFW) algorithm based on self-adaption principle and bimodal Gaussian function is proposed, which is built to optimize the PID controller by parameters tuning. Firstly, a compound index of optimization performance is formulated, and then the extremal optimization method of PID control system is proposed. Secondly, a PID parameters tuning model combined with AFW is built. At last, 5 typical transfer functions are simulated to obtain optimal parameters by AFW and contrast tuning method, such as Ziegler-Nichols method, Enhanced Fireworks (EFW) algorithm, and Particle Swarm Optimization (PSO). Simulation results show that AFW are effective and are easily implemented methods to solve PID control problems of different transfer functions.

1. Introduction

It is not only the simplest but also the most efficient control strategy to solve many real-world control problems by Proportional-Integral-Derivative (PID) control system [1]. Optimal parameters of PID controller may obtain high-quality performances such as high stability and satisfied transient response, but it is difficult to be tuned to control a complex system [2, 3]. In order to get better parameters of PID controller, more and more researchers pay attention to setting the parameters beyond cut-and-try or Ziegler-Nichols method, which always get low-quality performances and waste a lot of time.

The evolutionary algorithm (EA) principle is a hot research topic in optimization techniques and several of them have been adopted to solve these problems in the past decade, such as PSO (Particle Swarm Optimization) [4–6], GA (Genetic Algorithm) [7, 8], ACO (Ant Colony Optimization) [9], and BBO (Biogeography-Based Optimization) [10, 11].

A novel evolutionary algorithm named Fireworks Algorithm (FWA) is proposed and enhanced by Tan et al. [12, 13]. And then, Janecek and Tan applied Fireworks Algorithm to

nonnegative matrix [14]. Gao and Diao applied Fireworks Algorithm to digital filters design [15]. He et al. used Fireworks Algorithm for spam detection [16]. Du solved nonlinear equations with Fireworks Algorithm and compared it with Artificial Bee Colony (ABC) algorithm [17]. Although nearly all of these applications get effective results, its application to control systems is scarce and rarely found in the published literature [18].

As a consequence, this paper focuses on PID controller parameters tuning with optimization on Advanced Fireworks (AFW) algorithm in the attempt to obtain better performance. Firstly, compound index of PID optimization performance by analyzing PID controller and several traditional performance indices is built. Then, based on the simulation model combined with AFW, 5 typical transfer functions of control system were simulated to obtain optimal parameters by AFW and compared with Enhanced Fireworks (EFW) algorithm, Particle Swarm Optimization (PSO), and Genetic Algorithm (GA). Results show that AFWA is an effective and easily implemented method to solve PID control problems with different transfer functions.

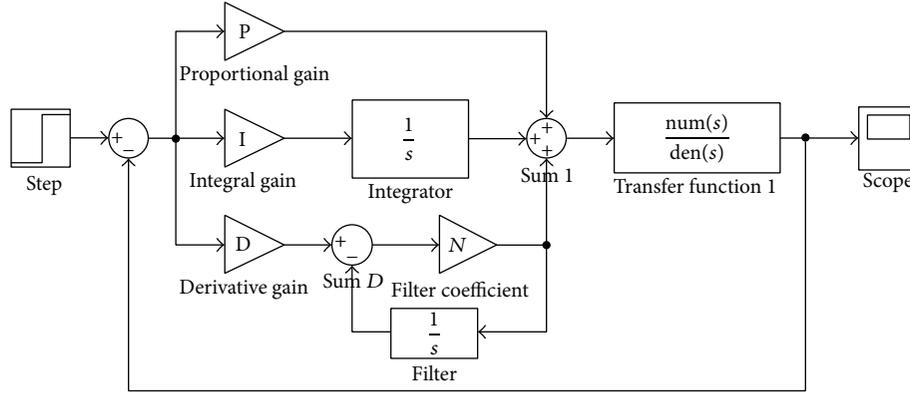


FIGURE 1: The structure of typical control system With PID controller.

2. PID Controller and Optimization Performance Index

2.1. PID Controller. A lot of practice showed that every PID controller parameter (proportional gain, integral gain, and derivative gain) has its own effectiveness to control system clearly, which is very easy to operate and also very effective. The structure of standard control system with PID controller is shown in Figure 1, and the transfer function in the Laplace Domain of the PID controller is described as follows:

$$L(s) = K_p + K_i \frac{1}{s} + K_d \frac{s}{s/N + 1}, \quad (1)$$

where K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, s is the complex number frequency, N is the filter coefficient, and $s/(s/N + 1)$ will be deduced to ideal differentiator s when N is large enough.

For a typical feedback control system, a continuous form of the PID controller is described as follows:

$$u(t) = K_p e(t) + K_i \int e(t) dx + K_d \frac{de(t)}{dt}, \quad (2)$$

where $u(t)$ is the out of the PID control signal and $e(t)$ is the system error, $e(t) = R(t) - y(t)$, which means the difference between the output value and the target value.

Furthermore, the discrete PID controller is written as follows:

$$u(k) = K_p e(k) + K_i T_s \sum_{j=1}^k e(j) + \frac{K_d}{T_s} [e(k) - e(k-1)], \quad (3)$$

where T_s is the sampling interval time.

2.2. Hybrid Index with Linear Weighted Method. There are several indices measuring the performance of the control system, such as the integral of the absolute value of error (IAE), the integral of time multiplied by the absolute value of error (ITAE), the integral of the square value of error (ISE), and the integral of time multiplied by the square of

error (ITSE). These indices did not fully express the control performances comprehensively, so we present another novel index which is more reasonable.

The index we proposed is a hybrid index ITUE which is created by the linear weighted method and combined ITSE output of PID controller and rate of system error into a single index by linear weighted method according to the importance of each index. It is defined as follows:

$$ITUE = \int \left\{ \omega_1 t e^2(t) + \omega_2 |u(t)| + \omega_3 \left| \frac{de(t)}{dt} \right| \right\} dt, \quad (4)$$

where ω_1 , ω_2 , and ω_3 are the weight value of ITSE, output of PID controller, and rate of system error, respectively. Besides ITSE, $u(t)$ is applied to avoid exporting too large control value that may be beyond the amplitude of PID controller in engineering and the rate of system error is used to avoid excessive rate of error which may cause the sensor delay in engineering.

Furthermore, the discrete performance index of control system is written as follows:

$$ITUE = \sum \left\{ \omega_1 e^2(k) + \omega_2 |T_s u(k)| + \omega_3 \left| \frac{e(k) - e(k-1)}{T_s} \right| \right\}. \quad (5)$$

3. PID Parameters Tuning Model Based on Advanced Fireworks Algorithm

In this section, the methodology for tuning the optimization of PID controllers is described. Firstly, a novel evolutionary algorithm named AFW is proposed, which is constituted of fireworks explosion, mutation explosion, and selection of explosion locations. And then, structure of PID parameters tuning model based on AFW is introduced.

3.1. Simulink Structure of Testing Performance Index on PID Control System. Simulink, developed by MathWorks, is a graphical programming environment for modeling, simulating, and analyzing control systems and offers tight integration

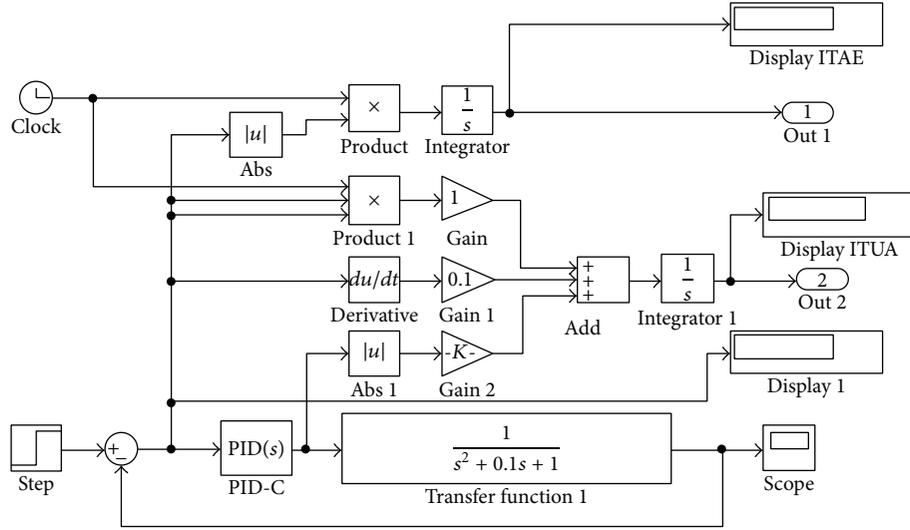


FIGURE 2: Simulink structure of testing performance index on PID control system.

with the rest of the MATLAB environment and can either drive MATLAB or be scripted from MATLAB. For Simulink is widely used in automatic control and digital signal processing simulation, testing performance index ITAE and ITUA on PID control system is built as in Figure 2.

3.2. Advanced Fireworks Algorithm. In order to improve computational efficiency of PID parameters tuning, Advanced Fireworks algorithm is proposed.

Advanced Fireworks space is made up by n firework devices and D -dimensional variable space (arguments space of optimization function), location of elements (fireworks, sparks, or mutation sparks), $X_i = (X_i^1, X_i^2, \dots, X_i^D)$, $i = 1, 2, \dots, n$, and gorgeous degree (objective function value) of elements $Y_i = f(X_i)$. Fireworks Algorithm is constituted by three steps (fireworks explosion, mutation explosion, and selection of explosion Locations).

In the beginning, 1st-generation location of firework i in dimension z is generated randomly as follows:

$$X_i^z = X_{\min}^z + (X_{\max}^z - X_{\min}^z) \times \text{rand}(0, 1), \quad (6)$$

where X_{\min}^z , X_{\max}^z are the lower and upper boundary of searching space in dimension z and $\text{rand}(0, 1)$ is displacement parameter generated from a standard uniform distribution on the open interval $(0, 1)$.

3.2.1. Fireworks Explosion. Fireworks explosion is proposed to generate sparks scattered all around the explosion location of father fireworks, which can be viewed as a searching process around a specific spot. High-quality fireworks generate numerous sparks which centralize the explosion center. That is to say, spot closed to the optimal location needs to be well searched. However, low-quality fireworks generate quite few sparks which are scattered in the space. In order to obtain splendid fireworks explosion, it is wise to set sparks number and explosion amplitude according to gorgeous degree (objective function value) of fireworks.

The number of sparks generated by firework i is expressed follows:

$$N_i = \text{round} \left(N_c \cdot \frac{f_{\max} - f(X_i) + \varepsilon}{\sum_{i=1}^N (f_{\max} - f(X_i)) + \varepsilon} \right), \quad (7)$$

where N_c is constant of total sparks number, f_{\max} is the maximum value of the objective function among the n fireworks, and ε is the machine epsilon. Due to the limitation of manufacturing process, number of the sparks generated by fireworks should be no more than N_{\max} and no less than N_{\min} : that is, N_i should be replaced by N_{\max} if N_i is bigger than N_{\max} , and N_i should be replaced by N_{\min} if N_i is smaller than N_{\min} .

The amplitude of fireworks explosion by firework i is expressed as follows:

$$A_i = A_c \cdot \frac{f(X_i) - f_{\min} + \varepsilon}{\sum_{i=1}^N (f(X_i) - f_{\min}) + \varepsilon}, \quad (8)$$

where A_c is constant of the maximum explosion amplitude and f_{\min} is the worst value of the objective function among n fireworks.

And then, N_i sparks are generated in the amplitude space randomly. z th dimension location of spark j generated from firework i is defined as follows:

$$X_{ij}^z = X_i^z + \delta_1^z \times A_i \times \text{rand}(-1, 1), \quad (9)$$

where X_i^z is the location of firework i in z th dimension coordinate, δ_1^z is changing factor which will be 0 or 1 randomly, and $\text{rand}(-1, 1)$ is displacement parameter generated from uniform distribution over $[-1, 1]$. δ_1^z denotes whether z th dimension location of spark j generated from firework i should be different from z th dimension location of firework i . Once X_{ij}^z is out of searching space in dimension z , X_{ij}^z will be regenerated as (9) until it is located in the searching space.

3.2.2. Mutation Explosion Based on Bimodal Gaussian Function. In order to keep the diversity of sparks, mutation explosion is proposed to generate mutation sparks which are able to escape from original location. Location of mutation spark is generated from and escape from local extremum area as much as possible. So we define z th dimension location of mutation spark j generated from firework i as follows:

$$Xm_{ij}^z = \delta_2^z \times Xf_i^z \times N_2, \quad (10)$$

where δ_2^z is changing factor of mutation explosion which will be 0 or 1 randomly and N_2 is displacement parameter generated from bimodal Gaussian function follows:

$$\begin{aligned} N_2 &= (X_{\min}^z + \sigma_1 * e) * \text{heaviside}(r_1 - r) \\ &\quad + (X_{\max}^z + \sigma_2 * e) * \text{heaviside}(r - r_1), \quad (11) \\ r &= \frac{Xf_i^z - X_{\min}^z}{X_{\max}^z - X_{\min}^z}, \end{aligned}$$

where σ_1 and σ_2 are the mean square error of bimodal Gaussian function, r_1 is a random number generated from $\text{rand}(0, 1)$, and $\text{heaviside}(x)$ is a MATLAB function which has the value 0 for $x < 0$, 1 for $x > 0$, and 0.5 for $x = 0$. Once Xm_{ij}^z is out of searching space in dimension z , Xm_{ij}^z is regenerated as (10) until it is located in the searching space.

3.2.3. Selection of Explosion Locations. After fireworks explosion and mutation explosion, we already get the location of fireworks, sparks, and mutation sparks. At the same time, the gorgeous degree (objective function value) can be calculated too. Then, n locations are selected to be fireworks in next generation. The element with best gorgeous degree (objective function value) is kept to be firework in next generation and the other $n - 1$ fireworks are selected based on their distance to other elements so as to keep diversity. Distance can be measured by any one of Manhattan distance, Euclidean distance, Angle-based distance, and so on. The probability of whether the element i is selected to be firework in next generation is expressed as follows:

$$\begin{aligned} p(i) &= \frac{R(i)}{\sum_{j \in K} R(j)}, \\ R(i) &= \sum_{j \in K} d(i, j) = \sum_{j \in K} \|X_i - X_j\|, \end{aligned} \quad (12)$$

where K is the set of all current locations of elements.

3.3. PID Parameters Tuning Model Based on AFW. In order to obtain the optimal performance of PID control system on different transfer function, it is necessary to get a set of PID controller parameters (optimal location of element) which promote minimum performance index ITUE/ITAE. Note that filter coefficient may vary under different physical implementation. Without loss of generality, we set filter coefficient N as a constant which is large enough. The model

of PID parameters tuning based on AFW is depicted as in Figure 3.

Step 1. Set parameters of AFW, number of fireworks n , maximum number of iterations N_{\max} , and number of mutation sparks.

Step 2. Initialize sets of PID parameters $x_i = [K_p, K_i, K_d]$, that is, location of fireworks randomly.

Step 3. Set off fireworks (sets of PID controller parameters) to generate sparks (new sets of PID controller parameters).

Step 4. Generate mutation sparks (the other new sets of PID controller parameters) according to mutation explosion.

Step 5. Check out gorgeous degree of all elements (fireworks, sparks, and mutation sparks). That is to say, simulate PID control system (shown as in Figure 2) with all set of PID controller parameters and then get corresponding performance index ITUE/ITAE.

Step 6. Select n elements to be fireworks in next generation. That is to say, select n sets of parameters to generate new set of parameters in simulation of next generation.

Step 7. If the performance of PID control system meets the requirement or the generation of searching evolution reaches the maximum number of iterations N_{\max} , fireworks with best gorgeous degree are chosen to be best solution of problem. That is to say, the set of PID parameters with best performance index is chosen to be the optimal PID parameters (K_p, K_i , and K_d). Otherwise go to Step 3.

4. Simulations and Analysis

4.1. Comparison Tuning Method: Ziegler-Nichols Method. In order to evaluate the PID controller parameters (K_p, K_i , and K_d) tuned by AFW, the Ziegler-Nichols tuning method (Z-N method) is introduced to be comparison tuning method. Z-N method is a heuristic method of tuning a PID controller which was developed by John G. Ziegler and Nathaniel B. Nichols.

Z-N method is performed by setting the integral gain K_i and derivative gain K_d as zero. The proportional gain K_p is then increased (from zero) until it reaches the ultimate gain K_u , at which the output of the control loop has stable and consistent oscillations. K_u and the oscillation period T_u are used to set the PID controller parameters (K_p, K_i , and K_d) as follows: $K_p = 0.60K_u$, $K_i = T_u/2$, and $K_d = T_u/8$ [19].

4.2. The Simulation Environment and Evaluation Standards. With the PID parameters tuning model based on AFW, simulation is carried out to test the performance of PID tuning index ITUE and optimizing capacity of AFW on the problem of PID controller parameter tuning. AFW and several compared evolutionary algorithms are applied to optimize the PID controller performance index (ITAE or ITUE) by parameters (the proportional gain K_p , integral gain

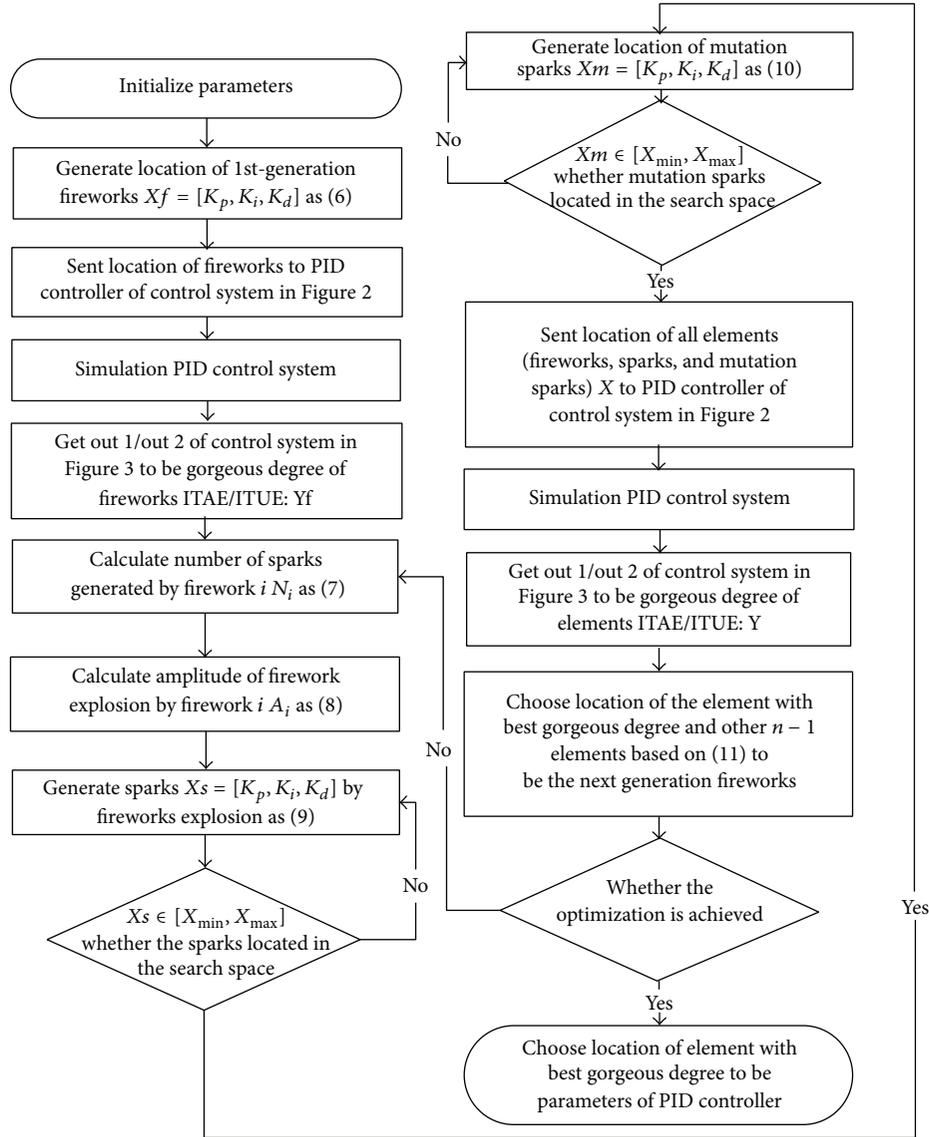


FIGURE 3: PID parameters tuning model based on Advanced Fireworks algorithm.

K_i , and differential gain K_d) tuning of PID controller with step response on 5 different typical transfer functions.

In order to fully compare the performance of different evolutionary algorithms and test performance of different PID controller performance index, it is essential to set similar simulation environment as soon as possible. Population size (partial size in PSO, fireworks size in AFW and EFW) was set as 50 and the maximum evaluation number as 100 for all algorithms on all functions. Simulation time of PID control system was set as 20 seconds, filter coefficient of PID controller was 100, and the weights values of the performance index ITUE were $w_1 = 1$, $w_2 = 0.1$, and $w_3 = 0.01$, respectively. Besides, other main parameters were chosen as in Table 1.

In order to evaluate the quality of the controller, step response is applied in this section. Step response is the

TABLE 1: Main parameters of evolutionary algorithms.

Method	Main parameters
AFW/EFW	$N_c = 50$, $N_{\min} = 2$, $N_{\max} = 40$, $A_c = 40$, mutation spark number $N_m = 5$
PSO	Inertia weight $w = 0.7298$, learning factor $c_1 = c_2 = 1.4946$, individual speed limitation is $[-0.5, 0.5]$

time behavior of the outputs of a general system when its inputs change from zero to one in a very short time. Then several quantities related to time behavior may be obtained, such as overshoot M_p and setting time t_s . Overshoot is the occurrence of a signal or function exceeding its target,

TABLE 2: Statistical results.

Function	Pa.	Z-N		AFW		EFW		PSO	
		ITAE	ITUE	ITAE	ITUE	ITAE	ITUE	ITAE	ITUE
(1) $\frac{1}{S^2 + 0.1S + 1}$	J	5.225	0.4536	0.0174	0.3646	0.0814	0.3649	0.0814	0.3672
	K_p	13.3021		0.6771	0.2594	30	0.2873	30	0.2791
	K_i	1.2164		7.5184	2.9116	30	3.1724	30	3.31
	K_d	12.0285		7.5108	2.9064	22.2751	3.1694	22.275	3.2908
	t_s	—		0.489	1.31	0.735	1.2	0.735	1.16
	M_p	4.36%		0%	0%	4.74%	0%	4.74%	0.0649%
(2) $\frac{300(s+30)}{S(s+10)(s+40)}$	J	0.05974	0.1008	1.917e-04	0.1046	0.0016	0.1043	0.0016	0.1223
	K_p	1.1504		23.1311	0.5909	8.6633	0.7817	8.7425	2.7446
	K_i	0.7772		0	0	30	0	30	9.2965
	K_d	0.0705		2.0809	0.052	0.36304	0.09986	0.3646	0.1514
	t_s	0.587		0.0643	0.288	0.155	0.342	0.154	0.375
	M_p	10.4%		49.7%	0%	25.2%	0%	25.3%	8.12%
(3) $\frac{250S+500}{S^3+12S^2+100S+10}$	J	23.29	7.8	0.0042	0.1211	0.0039	0.1190	0.0061	0.1226
	K_p	0.0271		21.6408	1.3742	30	1.1759	30	1.4559
	K_i	0.0043		2.1830	11.7829	24.2796	9.2061	30	13.6554
	K_d	0.0866		0.74008	0.12561	15.2678	0.105	3.1136	0.14221
	t_s	23.9		0.0938	0.67	0.0718	0.808	0.0697	0.59
	M_p	7.41%		43.5%	3.77%	75.7%	4.26%	57.3%	3.9%
(4) $\frac{S+5}{S^4+17S^3+60S^2+10S}$	J	13.19	2.938	0.1055	0.5265	0.4358	0.5265	0.4358	0.5265
	K_p	0.7484		5.2644	2.8763	30	2.8741	30	2.8886
	K_i	0.0020		0	0	0	0	0	0
	K_d	2.6791		29.9807	14.6236	30	14.623	30	14.6463
	t_s	21.1		2.4	1.07	2.85	2.41	2.85	2.39
	M_p	5.04%		0.784%	0.0349%	18.9%	0.778%	18.9%	0.804%
(5) $\frac{S+5}{S^4+17S^3+60S^2+5S+5}$	J	5.67	0.9095	0.1211	0.6971	1.1880	0.6974	1.1880	0.6985
	K_p	21.5801		1.7623	0.94238	30	0.95494	30	1.0341
	K_i	0.92192		2.508	1.2476	11.3251	1.2849	11.3253	1.3078
	K_d	30		29.5197	14.6483	30	15.0704	30	15.1394
	t_s	0.706		2.73	1.18	6.07	2.66	6.07	2.54
	M_p	10%		0%	0%	26.1%	0.0874%	26.1%	0.449%

which reflects the oscillation behavior of the system transient process. In particular, overshoot reflects the relative stability of the system. Settling time is defined as the time required for the response curve to reach and stay within a range of certain percentage (usually 5% or 2%) of the final value. t_s reflects overall rapidity of control system on step response.

4.3. Simulation and Analysis. Simulate the PID parameters tuning model based on AFWA with 5 typical transfer functions, respectively, in which filter coefficient is set as 100 according to testing benchmarks. Results of best ITAE, best ITUE, and their corresponding PID controller parameters are given as in Table 2. The best results between different methods are drawn in bold writing.

Based on PID control parameters tuned by Z-N method and optimal parameters (K_p , K_i , and K_d) with best ITAE (ITUA) performance optimized by AFW, EFW, and PSO, step response curves of PID control system on 5 typical transfer functions are drawn in Figure 4 (Figure 5), respectively.

It is shown that AFW almost has the best performance index value of 2 different performance indexes ITAE and ITUE in PID parameters tuning with transfer functions as (1) (2) (4) (5) or has little difference with the best performance index value but much better than the other evolutionary algorithms with transfer functions as (2) (3) according to Table 2.

Besides, curves with hybrid performance index ITUE get small overshoot and small rate of curve changing than cures with ITAE in each transfer function or each evolutionary algorithm according to Table 2, Figures 4 and 5.

5. Conclusion

In this paper, PID control system parameters tuning problem is transformed into a class of 3-dimensional parameter optimization problem based on performance indexes ITUE and ITAE. In order to design high-quality performances PID controller, an Advanced Fireworks (AFW) algorithm is

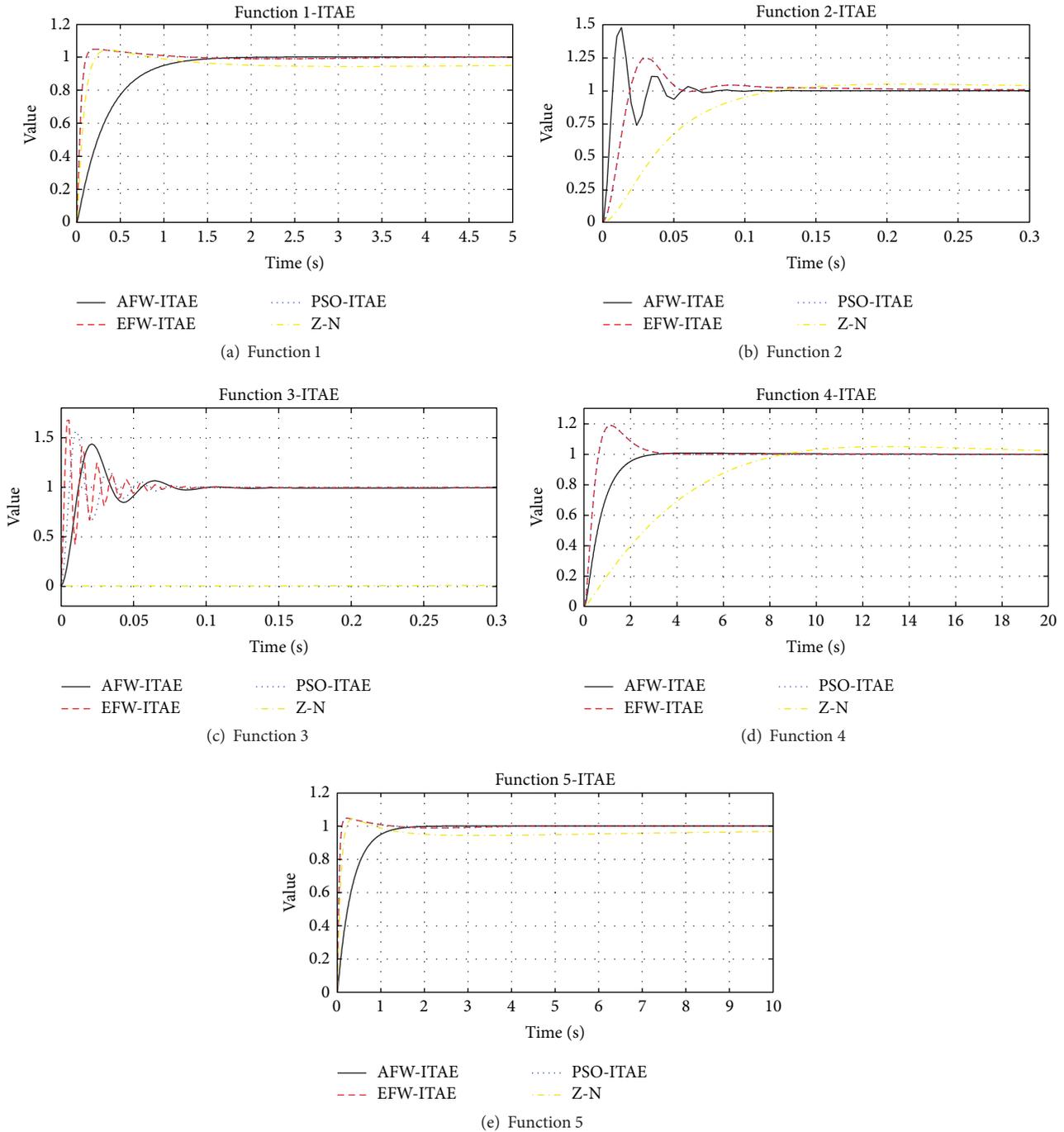


FIGURE 4: PID parameters tuning model based on Advanced Fireworks algorithm.

built to optimize the PID controller by parameters tuning. Extremal optimization method of PID control system was got by analyzing PID controller and building compound index of optimization performance. Then build a PID parameters tuning model combined with AFW. At last, 5 typical transfer functions of control system were simulated to obtain optimal parameters by AFW and were compared with Z-N tuning method, Enhanced Fireworks (EFW) algorithm, and Particle

Swarm Optimization (PSO). The simulation results show that AFW algorithms are effective and easily implemented methods to solve PID control problems of different transfer functions.

Competing Interests

The authors declare that they have no competing interests.

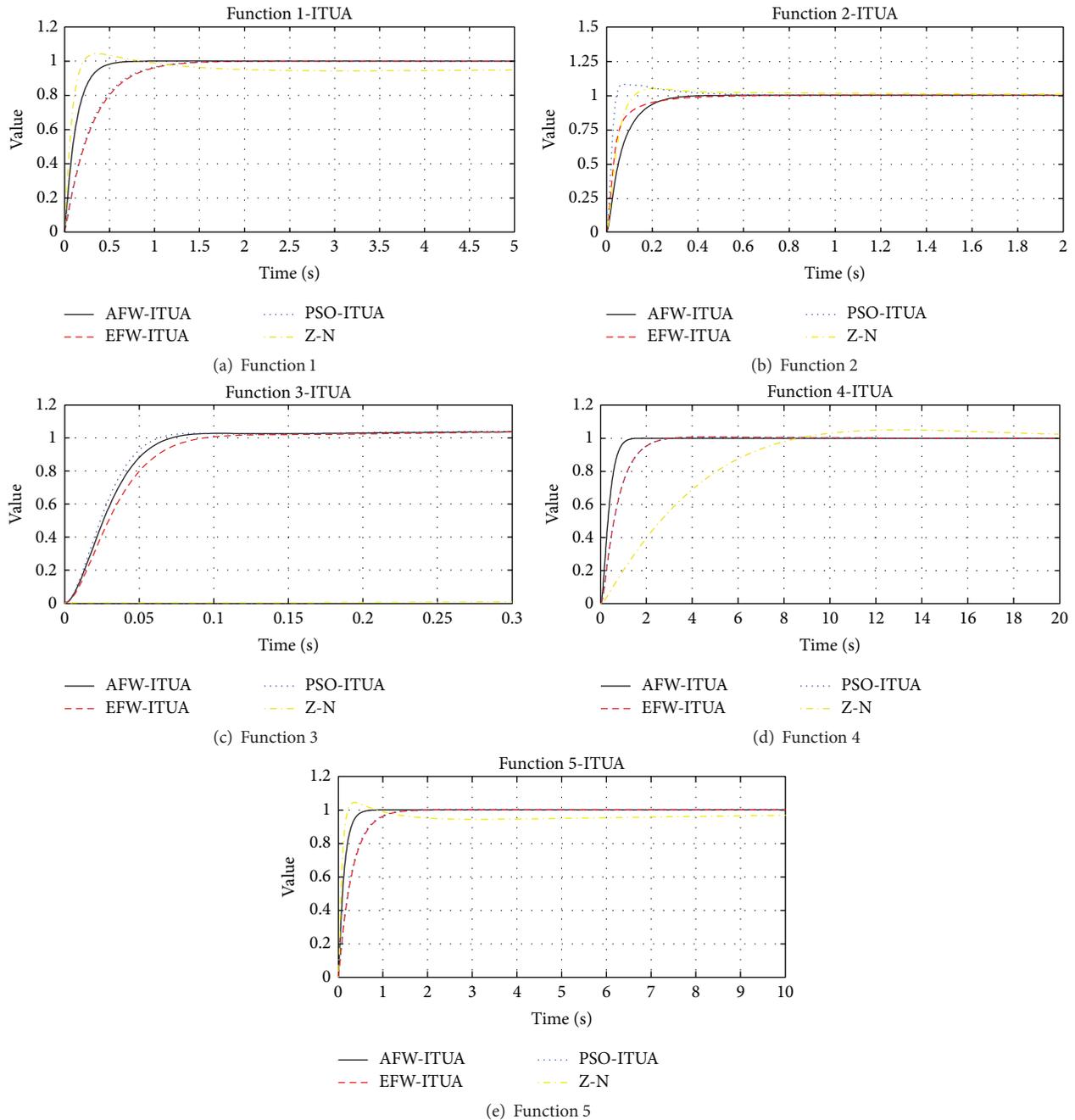


FIGURE 5: PID parameters tuning model based on Advanced Fireworks algorithm.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants nos. 71171199 and 61472443.

References

- [1] K. J. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163–1175, 2001.
- [2] A. I. Al-Odienat and A. A. Al-Lawama, "The advantages of PID fuzzy controllers over the conventional types," *American Journal of Applied Sciences*, vol. 5, no. 6, pp. 653–658, 2008.
- [3] W.-D. Chang and S.-P. Shih, "PID controller design of nonlinear systems using an improved particle swarm optimization approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3632–3639, 2010.
- [4] D.-F. Wang and P. Han, "Proportional-integral-derivative chaotic system control algorithm based on particle swarm optimization," *Acta Physica Sinica*, vol. 55, no. 4, pp. 1644–1650, 2006 (Chinese).
- [5] K. Khandani and A. A. Jalali, "PSO based optimal fractional PID controller design for an active magnetic bearing system," in *Proceedings of 18th Annual International Conference on Mechanical Engineering*, pp. 1–6, 2010.

- [6] G. Štimac, S. Braut, and R. Žigulić, “Comparative analysis of PSO algorithms for PID controller tuning,” *Chinese Journal of Mechanical Engineering*, vol. 27, no. 5, pp. 928–936, 2014.
- [7] L. Meng and S. Jiong, “Simulating study of adaptive GA-based PID parameter optimization for the control of superheated steam temperature,” *Proceedings of the CSEE*, vol. 22, no. 8, pp. 145–149, 2002 (Chinese).
- [8] P.-F. Niu, J. Zhang, and X.-P. Guan, “Research on genetic algorithm optimization based on PID control with two degrees of freedom controller for chaotic system,” *Acta Physica Sinica*, vol. 56, no. 7, pp. 3759–3765, 2007 (Chinese).
- [9] I. Chiha, N. Liouane, and P. Borne, “Tuning PID controller using multiobjective ant colony optimization,” *Applied Computational Intelligence and Soft Computing*, vol. 2012, Article ID 536326, 7 pages, 2012.
- [10] G.-Q. Zeng, K.-D. Lu, Y.-X. Dai et al., “Binary-coded extremal optimization for the design of PID controllers,” *Neurocomputing*, vol. 138, pp. 180–188, 2014.
- [11] M. M. Sayed, M. S. Saad, H. M. Emara, and E. E. Abou El-Zahab, “A novel method for PID tuning using a modified biogeography-based optimization algorithm,” in *Proceedings of the 24th Chinese Control and Decision Conference (CCDC '12)*, pp. 1642–1647, Taiyuan, China, May 2012.
- [12] Y. Tan and Y. Zhu, “Fireworks algorithm for optimization,” in *Advances in Swarm Intelligence: First International Conference, ICSI 2010, Beijing, China, June 12–15, 2010, Proceedings, Part I*, vol. 6145 of *Lecture Notes in Computer Science*, pp. 355–364, Springer, Berlin, Germany, 2010.
- [13] S. Zheng, A. Janecek, and Y. Tan, “Enhanced fireworks algorithm,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 62, pp. 2069–2077, Cancun, Mexico, June 2013.
- [14] A. Janecek and Y. Tan, “Swarm intelligence for non-negative matrix factorization,” *International Journal of Swarm Intelligence Research*, vol. 2, no. 4, pp. 12–34, 2011.
- [15] H. Gao and M. Diao, “Cultural firework algorithm and its application for digital filters design,” *International Journal of Modelling, Identification and Control*, vol. 14, no. 4, pp. 324–331, 2011.
- [16] W. He, G. Mi, and Y. Tan, “Parameter optimization of local-concentration model for spam detection by using fireworks algorithm,” in *Advances in Swarm Intelligence*, vol. 7928 of *Lecture Notes in Computer Science*, pp. 439–450, Springer, Berlin, Germany, 2013.
- [17] Z. Du, “Fireworks algorithm for solving nonlinear equation and system,” *Modern Computer*, vol. 6, no. 2, pp. 18–21, 2013.
- [18] G. J. Ni, P. Wang, and L. Liu, “PID optimization with regulation-based formulas and improved differential evolution algorithm,” *Advanced Engineering Forum*, vol. 6-7, pp. 261–266, 2012.
- [19] Microstar Laboratories, Ziegler-Nichols Tuning Rules for PID, <http://www.mstarlabs.com/control/znrule.html>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

