



UNIVERSITY OF LEEDS

This is a repository copy of *Knowledge-based particle swarm optimization for PID controller tuning*.

White Rose Research Online URL for this paper:  
<http://eprints.whiterose.ac.uk/156236/>

Version: Accepted Version

---

**Proceedings Paper:**

Chen, J, Omidvar, MN orcid.org/0000-0003-1944-4624, Azad, M et al. (1 more author) (2017) Knowledge-based particle swarm optimization for PID controller tuning. In: 2017 IEEE Congress on Evolutionary Computation (CEC). 2017 IEEE Congress on Evolutionary Computation (CEC), 05-08 Jun 2017, San Sebastian, Spain. IEEE , pp. 1819-1826. ISBN 9781509046010

<https://doi.org/10.1109/cec.2017.7969522>

---

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

**Reuse**

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.



[eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk)  
<https://eprints.whiterose.ac.uk/>

# Knowledge-based Particle Swarm Optimization for PID Controller Tuning

Junfeng Chen<sup>\*†</sup>, Mohammad Nabi Omidvar<sup>†</sup>, Morteza Azad<sup>†</sup>, Xin Yao<sup>‡†</sup>

<sup>\*</sup>College of IOT Engineering, Hohai University, Changzhou, China

<sup>†</sup>School of Computer Science, University of Birmingham, Birmingham, UK

<sup>‡</sup>Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

Email: chen-1997@163.com, m.omidvar@cs.bham.ac.uk, m.azad@cs.bham.ac.uk, xiny@sustc.edu.cn

**Abstract**—A proportional-integral-derivative (PID) controller is a control loop feedback mechanism widely employed in industrial control systems. The parameters tuning is a sticking point, having a great effect on the control performance of a PID system. There is no perfect rule for designing controllers, and finding an initial good guess for the parameters of a well-performing controller is difficult. In this paper, we develop a knowledge-based particle swarm optimization by incorporating the dynamic response information of PID into the optimizer. Prior knowledge not only empowers the particle swarm optimization algorithm to quickly identify the promising regions, but also helps the proposed algorithm to increase the solution precision in the limited running time. To benchmark the performance of the proposed algorithm, an electric pump drive and an automatic voltage regulator system are selected from industrial applications. The simulation results indicate that the proposed algorithm with a newly proposed performance index has a significant performance on both test cases and outperforms other algorithms in terms of overshoot, steady state error, and settling time.

**Index Terms**—Particle Swarm Optimization, PID Controller, Knowledge

## I. INTRODUCTION

The proportional-integral-derivative (PID) controller is a common control loop feedback mechanism in industrial control systems, the origin of which can be traced back to speed governor design in the 19th century. The parameters tuning is a hurdle in controller design, having a great effect on the performance of the industrial control systems, especially for those controlled plants with high order and time delays. Ziegler-Nichols (Z-N) [1] and Cohen-Coon [2] are the most commonly used methods for tuning PID controllers. Several intelligent methods, such as neural network, fuzzy system, and neural-fuzzy logic [3], have also been developed to optimize the parameters of PID controllers. Nature-inspired population-based metaheuristics [4] have also gained popularity for fine-tuning the parameters of PID controllers, due to their good performance and robustness.

Genetic algorithm (GA) [5] is a metaheuristic inspired by the process of natural selection and belongs to the larger class of evolutionary algorithms [6]. It was recommended as an important optimizer for nonlinear PID control systems [7]. Phillips et al. described the architecture of a helicopter

fuzzy logic controller and used GAs to discover rules for effective control of helicopters [8]. Herreros et al. regarded the design of a PID controller as a multi-objective problem and proposed a generic multi-objective method for designing PI and PID controllers [9]. After that, an alternative was provided by using GA and its flexibility is demonstrated by tuning an optimal PID in different cases: model errors, noisy input, Integral of Absolute Error (IAE) minimization, and following a reference models [10]. Chen et al. presented a distribution population-based GA, the searching capability of which was demonstrated by designing optimal parameters of PID controllers with several examples [11]. The crossover formula in GA was modified by Chang, who used this method to determine the gains of PID controller for multivariable processes [12]. Jan et al. proposed a robust PID control scheme for the permanent magnet synchronous motor drive by using a simple GA [13]. Later, the GA-based PID tuning method was extended to the electro-hydraulic servo actuator system [14]. Padhee et al. employed GA to carry out the design of fractional order PID controller, which is a special kind of PID controller whose derivative and integral order are fractional rather than integer [15]. A non-dominated sorting GA was designed and applied to PID-tuning for a robotic manipulator of two Degree-Of-Freedom (DOF) by Ayala. [16]. Boubertakh et al. employed a PSO method to design the decentralized PID controllers for the stabilization of a quadrotor [17].

Particle swarm optimization (PSO), intended for simulating social behavior, is another typical population-based metaheuristics [18]. Selvan et al. combined PSO with some special features and applied it to PID controller tuning [19]. Another design method which integrated the PSO algorithm with a new time-domain performance criterion was proposed to determine the parameters of PID controller [20]. Ko and Wu designed a fuzzy PID controllers for the multivariable seesaw systems and the PID gains and all other parameters were determined simultaneously [21]. Mukherjee and Ghoshal proved craziness based PSO was more robust than GA for performing the optimal PID gains even under various nominal operating conditions [22]. Zamani et al. employed PSO algorithm to determine the parameters  $\lambda$  (integral order) and  $\mu$  (derivative order) apart from the usual PID parameters for fractional order PID controller for an automatic voltage regulator [23]. Tarique

and Gabbar evaluated the feasibility of the use of PSO method for fine-tuning PID controllers in the steam turbine control system [24]. Boubertakh et al. suggested the use of PSO to solve the PID control design problem for angles and height stabilization of a quadrotor [25]. Malik et al. described the design of dynamic control system model with PID controller and the values of the controlling parameters were computed by using PSO [26]. Pano and Ouyang proposed a new fitness function based on the statistics of the contour error and applied PSO for control gain tuning of a position domain PID controller for a serial multi-DOF robotic manipulator [27]. The PSO-based PID tuning method was extended into the quadrotor's attitude and trajectory control in a cooperative aerial robot system [28].

Examples of population-based metaheuristics also include Ant Colony Optimization (ACO), Shuffled Frog Leaping Algorithm (SFLA), Bacterial Foraging Optimization (BFO) and so on. ACO algorithm was applied for optimizing the parameters in the design of a type of nonlinear PID controller by Duan [29], and then the grid-based ACO algorithm was designed for solving continual space flight optimization question [30]. Huynh introduced a modified SFLA into optimal tuning of PID gains for multivariable processes in the Wood-Berry distillation column system [31]. Atashpaz et al. described a socio-politically inspired colonial competitive algorithm, which was applied for the problem of designing a multivariable PID controller [32]. They also designed a centralized PID controller using covariance matrix adaptation evolution strategy for the same application of the multiple-input multiple-output (MIMO) systems [33]. Merheb and Noura explored a bio-inspired stochastic search algorithm for offline tuning of the PD controller of a quadrotor UAV by reference to ecosystem equilibrium [34]. Mohammed et al. described the PID controller for controlling attitude, Roll, Pitch and Yaw direction of quadrotor by using PSO, Bacterial Foraging Optimization (BFO) and the BF-PSO optimization [35]. Nagaraj et al. explored intelligent PID-tuning techniques like GA, Evolutionary programming and PSO for the armature controlled DC motor [36]. Iruthayarajan and Baskar analyzed performance comparison of several evolutionary algorithms on decoupled multivariable PI and PID controller, which mainly included real coded GA, modified PSO, covariance matrix adaptation evolution strategy and differential evolution (DE) [37]. Ghosal et al. reviewed the performances of PID tuning with different metaheuristic techniques, i.e., ACO, PSO, and BFO, as well as their advantages and disadvantages in proper tuning [38].

Hybridization is an important approach in search and optimization, having a great effect on global optimization performance, e.g., accuracy and convergence speed of metaheuristics. A number of publications document the benefits and great success of hybridizing metaheuristics with each other and/or with algorithms from other fields [39]. Korani presented an *E coli* algorithm for PID controller tuning based on a combination of the foraging behavior of BFO and PSO [40]. Kim suggested the hybrid system consisting of GA and

Bacterial Foraging, which was introduced into tuning for PID controller of AVR system with compared of GA, PSO, and GA-PSO hybrid algorithm [41]. Gharghory et al. presented an adaptive hybrid PSO by employing a mutation operator for local best particles and applied the proposed algorithm to self-tuning of PID controller in the ball and hoop system [42]. de Moura Oliveira and Cunha a teaching experiment in which PSO was blended with classical control techniques to design PID controllers [43]. Pai et al. developed PID and Sliding Mode Control methods for an Android-based quadrotor platform and parameters of optimum controller were built and implemented by using GA and Tabu Search [44]. For the purpose of fast tuning controller parameters, Fister et al. systematically investigated the performances of two reactive evolutionary algorithms (differential evolution and GA), and four reactive swarm intelligence-based algorithms (bat, hybrid bat, PSO and cuckoo search) in tuning the PID controller [45].

In view of the above-mentioned contributions, deliberate strategies and complex structures of metaheuristics are carefully designed to be served as a PID parameters optimizer. However, the tuning knowledge that has been identified and summarized in practice is seldom employed in the design of the PID controller. He et al. observed that the heuristic knowledge is useful for the Evolutionary Algorithms to find the optimal solutions for the node covering problem [46]. In this paper, the PSO algorithm is combined with the PID tuning rules, which help the algorithm to identify the promising regions quickly and increase the solution precision rapidly.

The rest of this paper is organized as follows. In Section II, the basic tuning theory is reviewed. In Section III, the relationship between PID parameters and response characteristics is analyzed and a knowledge-based particle swarm optimizer is proposed, followed by experimental simulation and result discussion on the industrial processes in Section IV. Finally, our concluding remarks are presented in Section V.

## II. PID TUNING THEORY

For simplicity, we focus on a PID controller in a closed-loop system using the schematic shown in Fig. 1 and expressed as Equation (1). The input  $r(t)$  is the desired process value or "set point", and the output  $y(t)$  is the actual output measured by detection equipment. The variable  $e(t) = r(t) - y(t)$  represents the tracking error, which will be sent to the PID controller, and the controller computes both the proportion, derivative and the integral of this error signal.

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}. \quad (1)$$

The control signal  $u(t)$  sent to the plant, is equal to the proportional gain ( $k_p$ ) times the magnitude of the error plus the integral gain ( $k_i$ ) times the integral of the error plus the derivative gain ( $k_d$ ) times the derivative of the error.

It is generally known that the dynamic performance of a control system is often measured by four major characteristics of the closed-loop step response, i.e., Rise Time ( $t_r$ ), Overshoot ( $\sigma\%$ ), Settling Time ( $t_s$ ) and Steady-state Error ( $e_{ss}$ ).

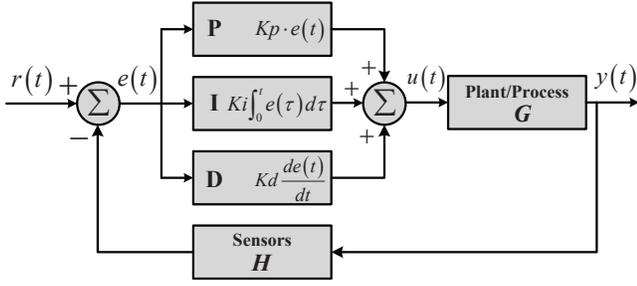


Fig. 1. A block diagram of a PID controller in a feedback loop.

More specifically,  $e_{ss}$  of the system under the step response is the difference between the input  $u(t)$  and the output  $y(t)$  when  $t \rightarrow \infty$ .  $t_r$  is the time it takes for the output signal  $y(t)$  to go from 10% to 90% of its steady-state value.  $t_s$  is time that  $y(t)$  enters and stays in the interval  $[y(\infty) - \Delta y, y(\infty) + \Delta y]$ , where the  $\Delta y$  is usually defined as either 2% or 5% of the steady-state value  $y(\infty)$ . The overshoot  $\sigma$  is defined using the following ratio:

$$\sigma = \frac{y_M - y(\infty)}{y(\infty)}, \quad (2)$$

where  $y_M$  is the peak value.

When we design a controller, it is expected to have a short starting time, high response speed, small overshoot and tracking error, and good robustness. Typical plants are selected from the practical applications which are System 1 formulated by Equations (3) and (4), and System 2 expressed as Equations (5) and (6).

$$\text{Plant 1 : } G_1(s) = \frac{0.5}{0.08s^2 + 0.68s + 1.45} e^{-0.2s}. \quad (3)$$

$$\text{Feedback 1 : } H_1(s) = 1. \quad (4)$$

$$\text{Plant 2 : } G_2(s) = \frac{10}{0.04s^3 + 0.54s^2 + 1.5s + 1}. \quad (5)$$

$$\text{Feedback 2 : } H_2(s) = \frac{1}{0.01s + 1}. \quad (6)$$

The controlled plant  $G_1$  in system 1 is a unit feedback model with time delay for an electric pump drive in a marine system [47]. The plant 2 is an automatic voltage regulator [20], which is often described as a four-order model ( $G_2$ ) with a non-unit feedback ( $H_2$ ). It is typical to solve the non-unit feedback problems by converting them to unit feedback using the forward-access model. However, in this paper we directly solve plants with non-unit feedback models. Fig. 2 shows the step responses of the plants without PID controllers. It can be seen roughly that the plant 1 has a large steady-state errors, while the plant 2 has a high overshoot and settling time.

### III. KNOWLEDGE-BASED PARTICLE SWARM OPTIMIZER

There is a tendency in the metaheuristics community to design sophisticated search strategies which are not problem-specific and blind to the acquired information and prior knowledge. The specific knowledge about the problem to

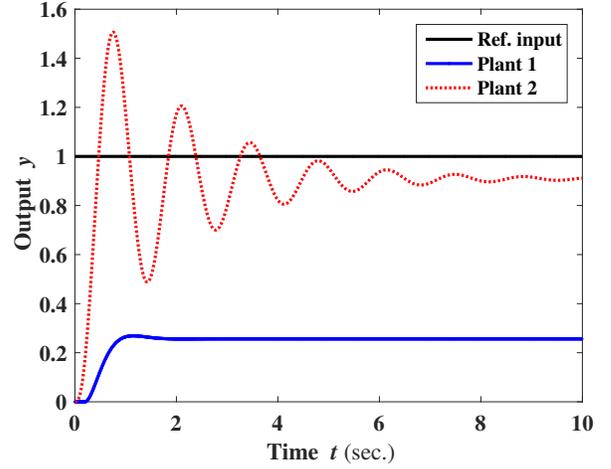


Fig. 2. The step responses of the plants without PID controller.

be solved has contributed to the development of population-based metaheuristic algorithms. The prior knowledge not only empowers metaheuristic algorithms to quickly identify the regions in the search space with high quality solutions, but it also helps metaheuristics to increase the solution precision in the limited running time. In other words, the information that we have already known about PID parameter tuning and optimizing will serve to guide further search and could be incorporated into heuristic algorithms, which is overlooked in the current metaheuristics development process.

For the specific problem of tuning PID controllers, the effects of increasing each of the control parameters  $k_p$ ,  $k_i$ , and  $k_d$  can be summarized into meaningful knowledge about the relationship between PID parameters and response characteristics for designing a PID controller (as shown in Table I). First, the proportional gain  $k_p$  can be used for decreasing the rise time. Second, the derivative gain  $k_d$  can regulate the overshoot and settling time. Third, the integral gain  $k_i$  contributes to eliminating the steady-state error.

TABLE I  
RELATIONSHIPS BETWEEN PID PARAMETERS AND RESPONSE CHARACTERISTICS [48]

Para.	Rise Time	Overshoot	Settling Time	Error
$k_p$	Decrease	Increase	Small change	Decrease
$k_i$	Decrease	Increase	Increase	Eliminate
$k_d$	Minor change	Decrease	Decrease	No Effect

In this paper, we discard the complex search strategies and combine useful dynamic response knowledge into the PSO algorithm. The basic PSO contains a swarm of particles whose movements are not only influenced by their local best known positions, but also guided toward the global best known positions in the search-space. After finding the two best values, the particles update their velocity and positions based on the following equations.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1^t (p_i^t - x_i^t) + c_2 r_2^t (g_i^t - x_i^t), \quad (7)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad (8)$$

where  $\omega$  is a parameter called inertia weight, the parameters  $c_1$  and  $c_2$  are called acceleration coefficients, and  $r_1^t$  and  $r_2^t$  are two  $n \times n$  diagonal matrices in which the entries are random numbers uniformly distributed in the interval  $[0, 1)$ . At each iteration, these matrices are regenerated.  $p_i^t$  and  $g_i^t$  are the personal best solution of the  $i$ -th particle and the global best solution ever found by any particle in the swarm, respectively.

Generally speaking, the value of inertia weight is linearly decreased over the generations to favor exploration in initial generations and exploitation in the later generations. The following equation is used to update the value of inertia weight.

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{\text{iter}_{\max}} \times \text{iter}, \quad (9)$$

where  $\omega_{\max}$  and  $\omega_{\min}$  are respectively the lower and upper boundaries of the inertia weight  $\omega$ . The argument  $\text{iter}_{\max}$  is the maximum number of iterations and the variable  $\text{iter}$  is the current iteration.

According to the above analyses, this paper proposes the knowledge-based Particle Swarm Optimization (KPSO) by a careful combination of the original PSO algorithm and the response characteristics. More specifically, the improvements mainly concern three aspects: the constitution of solution components, parameter setting based on prior knowledge, and evaluation function definition.

First, three controller parameters are defined to compose an individual  $\vec{x} = (k_p, k_d, k_i)$ , therefore, there are only three members in an individual. Each member is assigned as a real value. If there are  $n$  individuals in a population, then a population  $\mathbf{X}$  can be expressed as the following matrix form.

$$\mathbf{X} = \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_n \end{pmatrix} = \begin{pmatrix} k_{p1} & k_{d1} & k_{i1} \\ k_{p2} & k_{d2} & k_{i2} \\ \vdots & \vdots & \vdots \\ k_{pn} & k_{dn} & k_{in} \end{pmatrix}, \quad (10)$$

where the  $\vec{x}_j$  is a vector,  $j = 1, 2, \dots, n$ .

Second, some parameter settings are associated with the prior knowledge extracted from Table I. As the proportional gain  $k_p$  is related to the overshoot and steady-state error, the key parameter  $\omega$  in (7) is extended from a scalar to a vector, expressed as  $\vec{\omega} = (\omega_p, \omega_d, \omega_i)$ . The element  $\omega_p$  is fined as a nonlinear piece-wise function:

$$\omega_p = \begin{cases} \omega_{\max}, & \text{if } |\sigma - 1| \geq \epsilon \text{ or } |e_{ss}| \geq \tau \\ \frac{1}{\epsilon^2} [(\omega_{\max} - \omega_{\min})(\sigma - 1)^2 + \omega_{\min}] & \text{otherwise,} \end{cases} \quad (11)$$

where  $\epsilon$  and  $\tau$  are thresholds of overshoot and steady-state error, respectively. According to general definitions,  $\epsilon$  can be defined as 0.2 and  $\tau$  is 0.1.

Fig. 3 shows the inertia weight  $\omega_p$  according to the overshoot and steady-state error, where the  $\tau = 0.1$  and  $\epsilon = 0.2$ . The argument  $\omega_p$  increases non-linearly with  $(\sigma - 1)$  in the interval  $[-0.2, 0.2]$ .

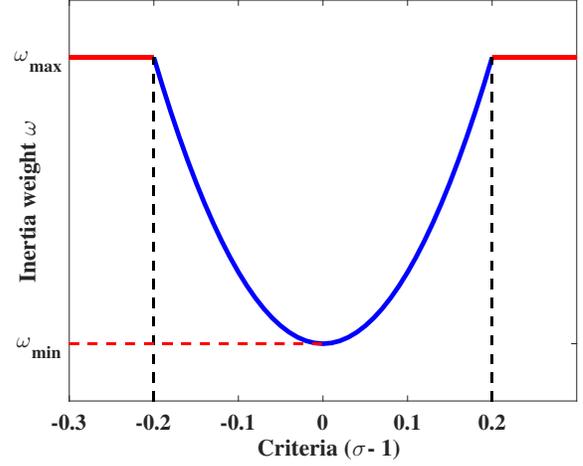


Fig. 3. The inertia weight  $\omega_p$ .

A minor modification is made to the position update equation (8) to meet the requirements of the special cases.

$$x_{j,p}^{t+1} = \begin{cases} x_{j,p}^t - |v_{j,p}^{t+1}| & e_{ss} \geq 0.1 \\ x_{j,p}^t + |v_{j,p}^{t+1}| & e_{ss} \leq -0.1, \end{cases} \quad (12)$$

where  $x_{j,p}^{t+1} = k_{p,j}^{t+1}$ ,  $j = 1, 2, \dots, n$ . The superscript  $t$  is indicated the variable  $\text{iter}$ .

Third, a more comprehensive performance index is designed not only based on integral error, but also based on the control input, rise time, settling time, etc. The four commonly used measures are Integral of Absolute Error (IAE), Integral of Time multiply Absolute Error (ITAE), Integral of Squared Error (ISE) and Integral of Time Multiply Squared Error (ITSE) [20], which are defined based on the integral error for a step set-point response:

$$\text{IAE} = \int_0^{\infty} |e(t)| dt \quad (13)$$

$$\text{ITAE} = \int_0^{\infty} t |e(t)| dt \quad (14)$$

$$\text{ISE} = \int_0^{\infty} e(t)^2 dt \quad (15)$$

$$\text{ITSE} = \int_0^{\infty} t e(t)^2 dt \quad (16)$$

All the measures require a fixed experiment to be performed on the system and the integrals are evaluated over a fixed time period (in theory to infinity, but usually until a time long enough for the responses to settle). These performance criteria in the frequency domain have advantages as well as disadvantages. For example, control systems specified to minimize ISE will tend to eliminate large errors quickly, but

will tolerate small errors persisting for a long period of time. Often this leads to fast responses, but with considerable low amplitude oscillation. IAE does not add weight to any of the errors, which tends to produce slower response than ISE optimal systems, but usually with less sustained oscillation. ITAE and ITSE integrate the absolute error or squared error weighted by the time and is integrated over time. The effect of this is to weight the errors. The downside of this is that ITAE/ITSE tuning also produces systems with sluggish initial response.

Another reason that these measures cannot be used in practical system comparisons is that they require a carefully controlled experiment. Although it may sometimes be possible to perform experiments on real plant, it is impossible to stop random disturbances affecting the process during an experiment.

In this paper, the time integral of absolute error is employed as a part of the performance index to achieve the satisfactory dynamic characteristics in the transition process and the squared controlled variable is adopted to avoid the high control outputs. Additionally, a weighted function is designed to eliminate the longtime adjusting phenomenon. Therefore, a new performance criterion (Integral of Absolute Error and Control Signal IAEU). is defined as follows:

$$\text{IAEU} = \int_0^{\infty} (|e(t)| + \theta_1 |u(t)|) dt + \theta_2 \sigma + \theta_3 (t_r + t_s), \quad (17)$$

where  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  are the weighting factor. The performance index can satisfy the designer's various requirements by changing the weighting factors. This paper presents a prior knowledge-based PSO-PID controller for searching the optimal or near optimal controller parameters for the PID control Systems. Real number encoding technique is employed to describe each individual. The searching procedures of the proposed prior knowledge-based PSO algorithm is described as Algorithm 1.

#### IV. SIMULATION EXPERIMENTS AND PERFORMANCE ANALYSIS

For better understanding the search behavior of the proposed KPSO, a set of simulation experiments are performed for the algorithm investigations, performance analysis, and comparisons. The parameters of each algorithm are provided and the empirical results are reported and discussed in this section.

##### A. Parameter Setting

Simulation Experiments are carried out on PC with Intel Core i7 4510U CPU with 2.6GHz and 8.00GB memory capacity. To verify the effectiveness of the proposed algorithm, standard GA and PSO algorithms are selected as the competitors. In all the following experiments, the population size is all set to 30, i.e.,  $n = 30$  and the maximum iteration is  $\text{iter}_{\max} = 50$ . The search scopes of individuals are specified as  $k_p \in [0, 10]$ ,  $k_d \in [0, 2]$ , and  $k_i \in [0, 10]$  for the electric pump drive system, and they are limited as  $k_p \in [0, 5]$ ,  $k_d \in [0, 2]$ ,

---

#### Algorithm 1: The Prior Knowledge-based PSO Algorithm.

---

```

1 Initialize the population uniformly and calculate the initial
  fitness.
2 while stopping criteria is not reached do
3   Test the closed-loop system stability and reinitialize the
    unstable particles.
4   Calculate  $t_r$ ,  $\sigma\%$ ,  $t_s$ , and  $e_{ss}$  for each particle.
5   Update  $\omega$  and  $\omega_p$  according to (9), and (11).
6   Update particles' velocity according to (7).
7   if  $v_{i,j}^{(t+1)} > v_j^{\max}$  then
8      $v_{i,j}^{(t+1)} = v_j^{\max}$ ;
9   if  $v_{i,j}^{(t+1)} < v_j^{\min}$  then
10     $v_{i,j}^{(t+1)} = v_j^{\min}$ ;
11  if  $e_{ss} \geq 0.2$  then
12     $x_{i,p}^{t+1} = x_{i,p}^t - |v_{i,p}^{t+1}|$ ;
13  else if  $e_{ss} \leq -0.2$  then
14     $x_{i,p}^{t+1} = x_{i,p}^t + |v_{i,p}^{t+1}|$ ;
15  else
16    Modify particles' position according to (8).
17  Use IAEU (17) to calculate the quality of particles.
18  Update the particles' personal best and the global best.
19 return the best solution found.

```

---

and  $k_i \in [0, 2]$  in the automatic voltage regulator. The other parameter set and strategies involved in these algorithm are listed as follows.

For the GA algorithm [49]:

- Elitist strategy.
- Non-linear ranking selection with the probability  $p_s = 0.10$ .
- Single point uniform crossover with the probability  $p_c = 0.75$ .
- Non-uniform mutation with the probability  $p_m = 0.10$ .

For the PSO algorithm [49]:

- Inertia weight  $\omega$  decreasing linearly over the iterations.
- $\omega_{\max} = 0.9$  and  $\omega_{\min} = 0.2$ .
- Acceleration coefficients  $c_1 = c_2 = 2$ .
- Maximum velocity  $v_{\max} = (x_{\max} - x_{\min})/5$ .

For the KPSO algorithm

- $\omega_{\max} = 0.5$  and  $\omega_{\min} = 0.2$ .
- Acceleration coefficients  $c_1 = c_2 = 2$
- Maximum velocity  $v_{\max} = (x_{\max} - x_{\min})/5$ .
- $\theta_1 = 0.1$ ,  $\theta_2 = 10$ ,  $\theta_3 = 2$ .

##### B. Performance Analysis

Each algorithm is tested 30 times independently to obtain reasonable statistical results for each testing system. Table II gives the best and median results of three gains, i.e., proportional gain  $k_p$ , derivative gain  $k_d$  and integral gain  $k_i$ . Table III shows the statistical results of performance indicators for system 1 and system 2 by using Ziegler-Nichols (Z-N), GA-PID, PSO-PID and KPSO-PID tuning algorithms with the proposed performance index IAEU. To check the significance of the results, a series of Wilcoxon rank-sum test

has been conducted with Holm  $p$ -value adjustment using a 95% confidence interval. The median of the best performing algorithm is shown in bold. Fig. 4 and 5 are the step responses of System 1 and System 2 optimized by GA, PSO, and KPSO with different performance criteria, respectively.

TABLE II  
COMPARISON IN THE BEST AND MEDIAN PARAMETERS OF DIFFERENT PID CONTROLLERS

System	para.	stats.	Z-N	GA	PSO	KPSO
System 1	$k_p$	median	5.2973	4.9941	4.8478	4.2002
		best	5.2973	4.8794	4.8611	3.2140
	$k_d$	median	0.6554	0.7184	0.6542	0.6568
		best	0.6554	0.6677	0.6179	0.3841
	$k_i$	median	10.276	6.5111	6.2057	6.5194
		best	10.276	6.9635	6.1794	6.7106
System 2	$k_p$	median	1.1550	0.7162	0.6622	0.6337
		best	1.1550	0.8861	0.6748	0.6248
	$k_d$	median	0.1422	0.5694	0.3452	0.2939
		best	0.1422	0.3158	0.2590	0.2190
	$k_i$	median	2.2514	0.7550	0.6543	0.4894
		best	2.2514	0.7984	0.6100	0.4600

From Table III, the PID controller by the Z-N tuning method is based on Nyquist frequency response. It can be seen that the Z-N PID controller has large overshoot and even poor stability in the control of plants. The KPSO-PID controller has an advantage on performance of overshoot, rising and settling time as compared to GA-PID and PSO-PID controllers, although it has a slightly long in rising and settling time in System 2.

From Figures 4(a), 4(b) and 5(a), 5(b), it is hard to judge which performance criteria is better than the other ones. For example, the GA-PID controller evaluated by IAE has an overshoot advantage on system 1, but do not perform well on system 2. The PSO-PID controller evaluated by ISE has a quick start, but there is a little defect in the overshoot. The GA-PID and PSO-PID controllers with IAEU performs slightly better than the controller with other performance criteria in overshoot and steady state error. From Figure 4(c) and 5(c), we can see that the KPSO-PID controller with the performance index IAEU has an impressive performance in overshoot and steady state error and settling time, despite its imperfection in the rising time. It is worth pointing out that these population-based metaheuristic are heuristic algorithms, which performance can be affected by various factors, i.e., search strategies, initial distribution of solutions, parameter values, and maximum iteration involved in algorithms.

## V. CONCLUSIONS

A PID controller is a generic controller widely used in over 90 percent of industrial control systems. The control equation involves three separate parameters (the Proportional, the Integral and Derivative terms), the tuning of which is still an open issue. In this paper, knowledge-based PSO algorithm is proposed based on the analysis of the relationships between

TABLE III  
STATISTICAL COMPARISON OF PERFORMANCE INDICATORS BY USING Z-N, GA-PID, PSO-PID AND KPSO-PID ALGORITHMS. THE HIGHLIGHTED ENTRIES ARE BASED ON A SERIES OF WILCOXON RANK-SUM TEST USING THE HOLM  $p$ -VALUE ADJUSTMENT.

System	para.	stats.	Z-N	GA	PSO	KPSO
System 1	$\sigma$	median	0.2333	0.0863	0.0512	<b>0.0132</b>
		best	0.2333	0.0553	0.0357	0.0000
		mean	0.2333	0.0751	0.0502	0.0171
		std.	0.0000	0.0441	0.0057	0.0180
	$t_r$	median	0.2200	<b>0.2050</b>	0.2350	0.2750
		best	0.2200	0.1700	0.2100	0.2000
		mean	0.2200	0.3583	0.2327	0.2970
		std.	0.0000	0.4792	0.0087	0.0843
	$t_s$	median	2.8000	1.7500	1.4500	<b>1.3000</b>
		best	2.8000	1.3400	1.3400	0.6900
		mean	2.8000	1.9433	1.4617	1.2503
		std.	0.0000	0.6324	0.1343	0.3638
$e_{ss}$	median	0.0000	0.0000	0.0000	0.0000	
	best	0.0000	0.0000	0.0000	0.0000	
	mean	0.0000	0.0000	0.0000	0.0000	
	std.	0.0000	0.0000	0.0000	0.0000	
System 2	$\sigma$	median	0.5888	0.0934	0.0226	<b>0.0076</b>
		best	0.5888	0.0317	0.0190	0.0001
		mean	0.5888	0.0976	0.0237	0.0095
		std.	0.0000	0.0388	0.0036	0.0079
	$t_r$	median	0.2200	<b>0.1400</b>	0.2150	0.2500
		best	0.2200	0.1000	0.1900	0.1900
		mean	0.2200	0.1430	0.2173	0.2540
		std.	0.0000	0.0314	0.0153	0.0340
	$t_s$	median	2.8000	1.1150	<b>0.8100</b>	0.8150
		best	2.8000	0.4200	0.3100	0.3500
		mean	2.8000	1.5057	0.7240	0.6730
		std.	0.0000	1.2106	0.2061	0.2779
$e_{ss}$	median	0.0000	0.0000	0.0000	0.0000	
	best	0.0000	0.0000	0.0000	0.0000	
	mean	0.0000	0.0000	0.0000	0.0000	
	std.	0.0000	0.0000	0.0000	0.0000	

PID parameters and response characteristics. The dynamic response information is fully utilized for the search in progress, including the monitoring of the stability for generating the new solution to replace the unstable ones; designing new update rules of inertia weight with respect to the values of  $\sigma\%$  and  $e_{ss}$ ; adding the position modification; and integrating the new time-domain performance criterion.

Through the simulation of the marine system and automatic voltage regulator system, the results show that the proposed controller can nearly perform an efficient search for the optimal PID controller parameters in comparison with GA-PID controller and PSO-PID controller. In addition, different performance estimation schemes are performed in order to verify the superiority of the proposed criterion. It is clear from the results that the proposed method can solve the searching and tuning problems of PID controller parameters more easily and quickly than the GA and PSO method.

## ACKNOWLEDGMENT

This work was partially supported by National Natural Science Foundation of China (grant no. 61403121), Funda-

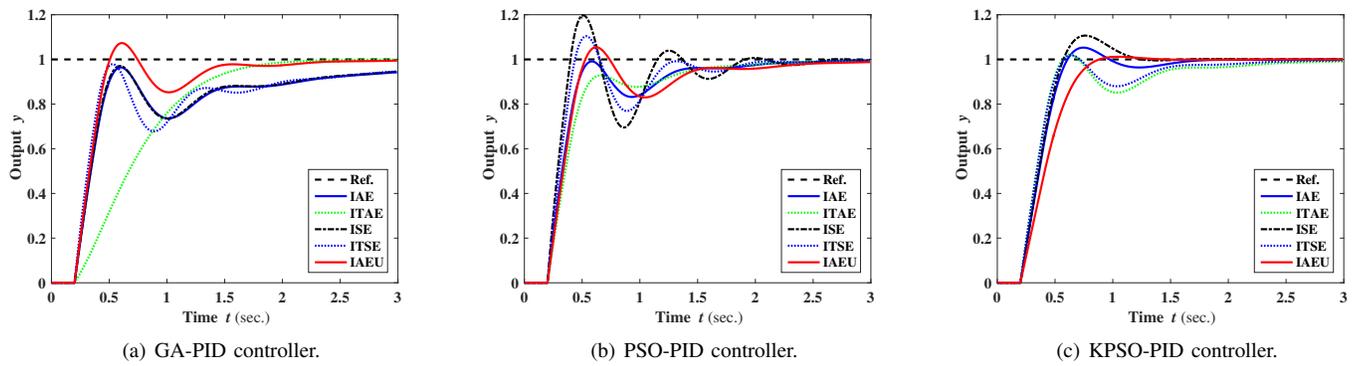


Fig. 4. The step responses of PID controller of System 1 optimized using GA, PSO, and KPSO with different performance criteria.

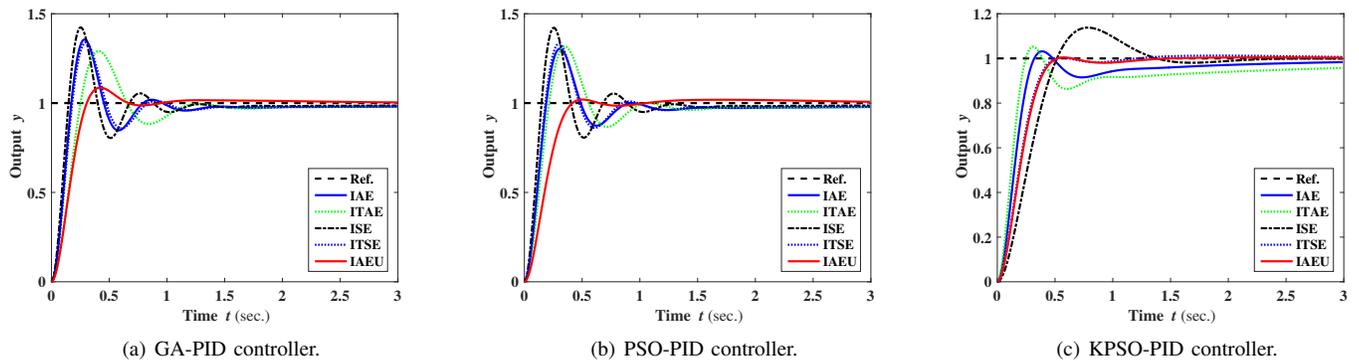


Fig. 5. The step responses of PID controller of System 2 optimized using GA, PSO, and KPSO with different performance criteria.

mental Research Funds for the Central Universities (grant no. 2015B20214), and EPSRC (grant nos. EP/K001523/1 and EP/J017515/1). Xin Yao was supported by a Royal Society Wolfson Research Merit Award.

## REFERENCES

- [1] P. Meshram and R. G. Kanojiya, "Tuning of pid controller using ziegler nichols method for speed control of dc motor," in *Proc. International Conference on Advances in Engineering, Science and Management (ICAESM'2012)*, Tamil Nadu, India, Mar. 30-31, 2012, pp. 117-122.
- [2] W. K. Ho, O. P. Gan, E. B. Tay and E. L. Ang, "Performance and gain and phase margins of well-known pid tuning formulas," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 4, pp. 473-477, 1996.
- [3] S.-Y. Chu and C.-C. Teng, "Tuning of pid controllers based on gain and phase margin specifications using fuzzy neural network," *Fuzzy sets and systems*, vol. 101, no. 1, pp. 21-30, 1999.
- [4] L. Bianchi, M. Dorigo, L. M. Gambardella and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing: an international journal*, vol. 8, no. 2, pp. 239-287, 2009.
- [5] M. Mitchell, *An introduction to genetic algorithms*, MIT press, 1998, pp. 2-10.
- [6] M. Patrascu, A. F. Stancu and F. Pop, "Helga: a heterogeneous encoding lifelike genetic algorithm for population evolution modeling and simulation," *Soft Computing*, vol. 18, no. 12, pp. 2565-2576, 2014.
- [7] P. Wang and D. P. Kwok, "Optimal design of PID process controllers based on genetic algorithms," *Control Engineering Practice*, vol. 2, no. 4, pp. 641-648, 1994.
- [8] C. Phillips, C. L. Karr and G. Walker, "Helicopter flight control with fuzzy logic and genetic algorithms," *Engineering Applications of Artificial Intelligence*, vol. 9, no. 2, pp. 175-184, 1996.
- [9] A. Herreros, E. Baeyens, and J. R. Perán, "Design of PID-type controllers using multiobjective genetic algorithms," *ISA Transactions*, vol. 41, no. 4, pp. 457-472, 2002.
- [10] J. M. Herrero, X. Blasco, M. Martínez and J. V. Salcedo, "Optimal PID tuning with genetic algorithms for non-linear process models," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 31-36, 2002.
- [11] Q.-G. Chen, N. Wang and S.-F. Huang, "The distribution population-based genetic algorithm for parameter optimization PID controller," *Acta Automatica Sinica*, vol. 31, no. 4, pp. 646-650, 2005.
- [12] W.-D. Chang, "A multi-crossover genetic approach to multivariable PID controllers tuning," *Expert Systems with Applications*, vol. 33, no. 3, pp. 620-626, 2007.
- [13] R.-M. Jan, C.-S. Tseng and R.-J. Liu, "Robust PID control design for permanent magnet synchronous motor: A genetic approach," *Electric Power Systems Research*, vol. 78, no. 7, pp. 1161-1168, 2008.
- [14] K. M. Elbayomy, Z. X. Jiao and H. Q. Zhang, "PID controller optimization by GA and its performances on the electro-hydraulic servo control system," *Chinese Journal of Aeronautics*, vol. 21, no. 4, pp. 378-384, 2008.
- [15] S. Padhee, A. Gautam, Y. Singh and G. Kaur, "A novel evolutionary tuning method for fractional order PID controller," *International Journal of Soft Computing and Engineering*, vol. 1, no. 3, pp. 1-9, 2011.
- [16] H. V. H. Ayala and L. dos Santos Coelho, "Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8968-8974, 2012.
- [17] H. Boubertakh, S. Bencharef and S. Labiod, "PSO-based PID control design for the stabilization of a quadrotor," in *Proc. 3rd International Conference on Systems and Control (ICSC)*, Algiers, Algeria, Oct. 2013, pp. 514-517.
- [18] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. 1998 IEEE International Conference on Evolutionary Computation (CEC)*, Anchorage, Alaska, May 1998, pp. 69-73.
- [19] S. E. Selvan, S. Subramanian and S. T. Solomon, "Novel technique for PID tuning by particle swarm optimization," in *Proc. 7th Annual Swarm Users/Researchers Conference (SwarmFest 2003)*, Indiana, USA, Apr. 2003, pp. 1-10.

- [20] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE transactions on energy conversion*, vol. 19, no. 2, pp. 384-391, 2004.
- [21] C.-N. Ko and C.-J. Wu, "A PSO-tuning method for design of fuzzy PID controllers," *Journal of Vibration and Control*, vol. 14, no. 3, pp. 375-395, 2008.
- [22] V. Mukherjee and S. Ghoshal, "Intelligent particle swarm optimized fuzzy pid controller for avr system," *Electric Power Systems Research*, vol. 77, no. 12, pp.1689-1698, 2007.
- [23] M. Zamani, M. Karimi-Ghartemani, N. Sadati and M. Parniani, "Design of a fractional order PID controller for an AVR using particle swarm optimization," *Control Engineering Practice*, vol. 17, no. 12, pp. 1380-1387, 2009.
- [24] A. Tarique and H. A. Gabbar, "Particle swarm optimization (PSO) based turbine control," *Intelligent Control and Automation*, vol. 4, no. 2, pp. 126-137, 2013.
- [25] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati and S. M. Smailzadeh, "Attitude control of a quadrotor with optimized PID controller," *Intelligent Control and Automation*, vol. 4, no. 3, pp. 335-342, 2013.
- [26] S. Malik, P. Dutta, S. Chakrabarti and A. Barman, "Parameter estimation of a PID controller using particle swarm optimization algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 3, pp. 5827-5830, 2014.
- [27] V. Pano and P. Ouyang, "Gain tuning of position domain PID control using particle swarm optimization," *Robotica*, vol. 34, no. 6, pp. 1351-1366, 2016.
- [28] J. Estevez, J. M. Lopez-Guede and M. Graña, "Particle swarm optimization quadrotor control for cooperative aerial transportation of deformable linear objects," *Cybernetics and Systems*, vol. 47, no. 1-2, pp. 4-16, 2016.
- [29] H.-b. Duan, D.-b. Wang and X.-f. Yu, "Novel approach to nonlinear pid parameter optimization using ant colony optimization algorithm," *Journal of Bionic Engineering*, vol. 3, no. 2, pp. 73-78, 2006.
- [30] H. Duan, S. Liu, D. Wang and X. Yu, "Design and realization of hybrid ACO-based PID and LuGre friction compensation controller for three degree-of-freedom high precision flight simulator," *Simulation Modelling Practice and Theory*, vol. 17, no. 6, pp. 1160-1169, 2009.
- [31] T.-H. Huynh, "A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers," in *Proc. the IEEE International Conference on Industrial Technology (ICIT 2008)*, Chengdu, China, Apr. 2008, pp. 1-6.
- [32] E. Atashpaz Gargari, F. Hashemzadeh, R. Rajabioun and C. Lucas, "Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process," *International Journal of Intelligent Computing and Cybernetics*, vol. 1, no. 3, pp. 337-355, 2008.
- [33] M. W. Iruthayarajan, S. Baskar, "Covariance matrix adaptation evolution strategy based design of centralized pid controller," *Expert systems with Applications*, vol. 37, no. 8, pp. 5775-5781, 2010.
- [34] A. R. Merheb and H. Noura, "Novel bioinspired stochastic tuning of a quadrotor PD controller," in *Proc. the 2nd Australian Control Conference (AUCC)*, Sydney, Australia, Nov. 15-16, 2012, pp. 227-232.
- [35] M. J. Mohammed, M. T. Rashid and A. A. Ali, "Design optimal PID controller for quad rotor system," *International Journal of Computer Applications*, vol. 106, no. 3, pp. 15-20, 2014.
- [36] B. Nagaraj, S. Subha and B. Rampriya, "Tuning algorithms for PID controller using soft computing techniques," *International Journal of Computer Science and Network Security*, vol. 8, no. 4, pp. 278-281, 2008.
- [37] M. W. Iruthayarajan and S. Baskar, "Evolutionary algorithms based design of multivariable PID controller," *Expert Systems with applications*, vol. 36, no. 5, pp. 9159-9167, 2009.
- [38] S. Ghosal, R. Darbar, B. Neogi, A. Das and D. N. Tibarewala, "Application of swarm intelligence computation techniques in PID controller tuning: A review," in *Proc. of the International Conference on Information Systems Design and Intelligent Applications (INDIA 2012)*, Visakhapatnam, India, Jan. 2012, pp. 195-208.
- [39] G. R. Raidl, "A unified view on hybrid metaheuristics," in *Proc. of the Third International Workshop on Hybrid Metaheuristics*, Gran Canaria, Spain, Oct. 13-14, 2006, pp. 1-12.
- [40] W. M. Korani, H. T. Dorrah and H. M. Emara, "Bacterial foraging oriented by particle swarm optimization strategy for PID tuning," in *Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Daejeon, Korea, Dec. 15-18, 2009, pp. 445-450.
- [41] D. H. Kim, "Hybrid GACBF based intelligent PID controller tuning for AVR system," *Applied Soft Computing*, vol. 11, no. 1, pp. 11-22, 2011.
- [42] S. M. Gharghory and H. A. Kamal, "Optimal tuning of PID controller using adaptive hybrid particle swarm optimization algorithm," *International Journal of Computers Communications & Control*, vol. 7, no. 1, pp. 101-114, 2012.
- [43] P. de Moura Oliveira and J. B. Cunha, "Blending Artificial Intelligence into PID Controller Design: A Biomedical Engineering Experiment," *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 366-371, 2016.
- [44] N.-S. Pai, W.-C. Li, M.-H. Chou and P.-Y. Chen, "Flight control for a quadrotor of attitude control based on android system and using optimal-tuning design," *Computers & Electrical Engineering*, vol. 54, pp. 170-184, 2016.
- [45] D. Fister, I. Fister and R. Šafarič, "Parameter tuning of PID controller with reactive nature-inspired algorithms," *Robotics and Autonomous Systems*, vol. 84, pp. 64-75, 2016.
- [46] J. He, X. Yao and J. Li, "A comparative study of three evolutionary algorithms incorporating different amounts of domain knowledge for node covering problem," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 2, pp. 266-271, 2005.
- [47] T. Roskilly and R. Mikalsen, *Marine systems identification, modeling and control*, Butterworth-Heinemann, 2015, pp. 88-89.
- [48] J. Zhong, *PID controller tuning: a short tutorial (class lesson)*, Purdue University, 2006.
- [49] J. F. Chen, Z. W. Ren and X. N. Fan, "Particle swarm optimization with adaptive mutation and its application research in tuning of PID parameters," in *Proc. 1st IEEE International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA)*, Harbin, China, Jan. 19-21, 2006, pp.990-994.