IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 6, 2020, accepted March 4, 2020, date of publication March 10, 2020, date of current version March 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979810

PID Controller Autotuning Design by a Deterministic Q-SLP Algorithm

JIRAPUN PONGFAI[1], XIAOJIE SU[2], HUIYAN ZHANG[3], AND WUDHICHAI ASSAWINCHAICHOTE[1]

[1]Department of Electronic and Telecommunication Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand
[2]College of Automation, Chongqing University, Chongqing 400044, China
[3]National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing 400067, China

Corresponding author: Wudhichai Assawinchaichote (wudhichai.asa@kmutt.ac.th)

This work was supported in part by the Petchra Pra Jom Klao Scholarship and in part by the Department of Electronic and Telecommunication Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi.

ABSTRACT The proportional integral and derivative (PID) controller is extensively applied in many applications. However, three parameters must be properly adjusted to ensure effective performance of the control system: the proportional gain (K_P), integral gain (K_I) and derivative gain (K_D). Therefore, the aim of this paper is to optimize and improve the stability, convergence and performance in autotuning the PID parameter by using a deterministic Q-SLP algorithm. The proposed method is a combination of the swarm learning process (SLP) algorithm and Q-learning algorithm. The Q-learning algorithm is applied to optimize the weight updating of the SLP algorithm based on the new deterministic rule and closed-loop stabilization of the learning rate. To validate the global optimization of the deterministic rule, it is proven based on the Bellman equation, and the stability of the learning process is proven with respect to the Lyapunov stability theorem. Additionally, to demonstrate the superiority of the performance and convergence in autotuning the PID parameter, simulation results of the proposed method are compared with those based on the central position control (CPC) system using the traditional SLP algorithm, the whale optimization algorithm (WOA) and improved particle swarm optimization (IPSO). The comparison shows that the proposed method can provide results superior to those of the other algorithms with respect to both performance indices and convergence.

INDEX TERMS Autotuning gain, central position control system, Q-learning algorithm, PID controller, swarm learning process algorithm, optimal control.

I. INTRODUCTION

Nowadays, PID controllers are widely applied and represent the most preferred choice of controller in many applications, such as power plants, industrial and mechanical systems, robotics [1]–[4], wind turbines [5], passive optical networks [6], [7], load frequency control (LFC) systems [8], [9], hydraulic turbine regulation systems [3], and radial active magnetic bearings [4], because they provide excellent performance, reliability, and robustness and are characterized by flexibility, low cost,a simple structure and ease of design [3], [10]–[13]. To obtain good closed-loop performance of PID controllers, appropriately adjusting three parameters, the proportional gain (K_P), integral gain (K_I) and derivative gain (K_D), is integral to system control [4], [10]. In 1942, a method of tuning the PID parameter was proposed by Ziglor-Nichlor [14]. After that, many methods were proposed, for example, the Cohen-Coon, phase and gain margin methods. These methods adjust the parameter based on the experience of the designer and require time for tuning. In practice, the system control is dynamic, and these approaches are ineffective for high order system modeling [7], [15], [16]. Recently, optimal tuning of the PID parameter by applying artificial intelligence (AI) has been implemented. This approach involves continuous tuning of the parameter based on the dynamic system to obtain the best system response by minimizing selection of the performance indices [17], [18].

The associate editor coordinating the review of this manuscript and approving it for publication was Huiping Li.

At present, many AI methods have been proposed to autotune the PID parameter, such as fuzzy logic [2], [10], [19], [20], neural networks (NNs) [1], [7], [21], particle swarm optimization (PSO) algorithms [6], [22], [23], hybrid firefly (FA) and pattern search [8], the ant lion optimization (ALO) algorithm [24], the whale optimization algorithm (WOA) [25], cuckoo search (CS) [10], [26], bacterial foraging optimization [27], genetic algorithms [28], the cosine algorithm [29], the bat algorithm [12], ant colony optimization (ACO) [13], [30], differential evolution (DE) [31], World Cup optimization (WCO) [32], evaluation algorithms (EAs) [33], [34], gray wolf optimization (GWO) [35], nature-inspired algorithms [17], chaotic invasive weed optimization [36], [37], flower pollination algorithm (FPA) [38] and firefly algorithm (FFA) [39]. Although many AI methods have been proposed to autotune the PID parameter, the challenges of long execution time and convergence persist. Therefore, [14] proposed the novel algorithm called the swarm learning process algorithm to improve the convergence and performance in autotuning the PID parameter. This algorithm is motivated by student learning in the classroom and applies the concepts of the swarm algorithm and learning algorithm. Achieving efficiency of the SLP algorithm requires adjusting the weight according to the behavior of the system. Conventionally, it is adjusted by a random process. Reinforcement learning (RL) is a method that solves the searching problem via interaction between an agent and the environment without needing an exact model of the environment [40], [41]. The agent receives the previous result of the learning environment in the form of a reward and learns until achieving the goal of learning. The Q-learning algorithm is an RL method that is widely applied in various applications, i.e., industrial control, robotics, time prediction, signal control, etc., because of the search rapidity and high convergence.

Due to the limitations of the SLP algorithm and the usefulness of the Q-learning algorithm, this paper applies the Q-learning algorithm to adjust the weight of the SLP algorithm. Nevertheless, the advantages of the Q-learning algorithm depend on the rule of updating the learning state [40], [42]–[44]. This paper proposes a new deterministic Q-learning algorithm for improving the stability and convergence performance in autotuning the PID parameter. Additionally, to improve the stability in adjusting the weight of the SLP algorithm based on the Q-learning algorithm, this paper proposes closed-loop stabilization of the learning rate, applied in the process of the Q-learning algorithm. The sufficient condition of the closed-loop stability is proven according to the Lyapunov stability theorem. Furthermore, the optimization of the proposed method is proven based on the Bellman equation. Finally, to show the superiority of the proposed method in terms of convergence and performance, a comparison with the simulation results of the traditional SLP algorithm [14], the WOA [25] and IPSO [45] based on the central position control (CPC) system is provided. Therefore, the contributions of this paper can be summarized as follows:

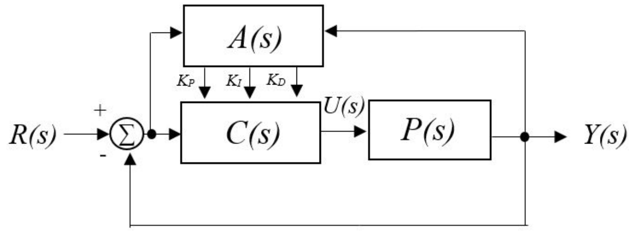

FIGURE 1. The structure of autotuning the PID controller.

1. The deterministic Q-SLP algorithm is proposed to autotune the PID parameter. This algorithm is a combination of the Q-learning algorithm and SLP algorithm. The Q-learning algorithm is applied to adjust the weight of the SLP algorithm.

2. The deterministic Q-SLP algorithm improves the stability, convergence and performance in autotuning the PID parameter.

3. A new rule for updating the process of the Q-learning algorithm is proposed, and its global optimization is proven based on the Bellman equation.

4. Closed-loop stabilization of the learning rate is proposed to improve the stability in adjusting the weight of the SLP algorithm based on the Q-learning algorithm. It is proven with respect to the Lyapunov stability theorem.

5. The superiority of the performance and convergence time are verified by comparison with the simulation results of the traditional SLP algorithm [14], the WOA [27] and IPSO [47] based on the CPC system.

This paper is organized into 6 sections: Section 2 presents the PID controller and objective design. The deterministic Q-SLP algorithm is presented in section 3. Section 4 explains the convergence analysis of the deterministic Q-SLP algorithm. Section 5 presents the simulation results and a discussion, and section 6 provides the conclusion.

II. PID CONTROLLER AND OBJECTIVE DESIGN

Generally, the structure of a PID controller combines three control parameters, i.e., K_P, K_I and K_D. Each parameter affects a different response of the system: K_P reduces the error, K_I increases the speed of response and reduces the error when the operation of the system is changed but overshoot occurs, and K_D reduces the overshoot from K_I and improves the stability of the system [7], [26]. The common closed-loop diagram of autotuning the PID controller is shown in Figure 1.

From Figure 1, the common transfer function of the PID controller ($C(s)$) is defined as follows:

$$C(s) = K_P + \frac{K_I}{s} + K_D s \tag{1}$$

Therefore, the output of the controller ($U(s)$) can be defined as follows:

$$U(s) = C(s)(R(s) - Y(s)) \tag{2}$$

where $P(s)$ is the transfer function of process control, $R(s)$ is the transfer function of reference input and $Y(s)$ is a transfer function of system response. $A(s)$ is the autotuning method that this paper proposes, the deterministic Q-SLP algorithm.

Because the primary objective of the optimizing algorithm is to find the proper value for achieving the objective function [29], the performance index of the optimal PID controller is significant. The performance indices usually considered in PID controller design are the integral of the absolute error (IAE), integral of the time multiplied squared error ($ITSE$), integral of the time multiplied absolute error ($ITAE$), mean of the squared error (MSE) and integral of the squared error (ISE) [12], [42]. The definition for each of the performance indices is as follows:

$$IAE = \int_0^\tau |e(t)|dt \tag{3}$$

$$ITSE = \int_0^\tau te(t)^2 dt \tag{4}$$

$$ITAE = \int_0^\tau t|e(t)|dt \tag{5}$$

$$MSE = \frac{1}{t}\int_0^\tau (e(t))^2 dt \tag{6}$$

$$ISE = \int_0^\tau e(t)^2 dt \tag{7}$$

where $e(t)$ is the error in the time domain.

Although the above performance indices are generally applied as the criteria of designing PID controllers because evaluation is based on the frequency domain, they have many disadvantages such as the response result based on IAE and ISE having a long settling time and the derivation process of $ITSE$ being complex [46], [47]. Therefore, [46], [47] proposed the new performance index given as Equation 8. It can be evaluated in the time domain by considering the transient response parameters, namely, the maximum overshoot (M_p), settling time (t_s), rise time (t_r) and steady state error (e_{ss}). This paper applies it as the criterion of autotuning the PID parameter and the basis of closed-loop stabilization of the learning rate.

$$J(t) = (1 - e^{-\beta})(m_p + e_{ss}) + e^{-\beta}(t_s - t_r) \tag{8}$$

where β is the weighting factor, which is set by the designer. If it is set to greater than 0.7, then the maximum overshoot and steady-state error are reduced. If it is set to less than 0.7, then the rise time and settling time are reduced.

III. DETERMINISTIC Q-SLP ALGORITHM

A. SLP ALGORITHM

The SLP algorithm was proposed by [14]. It applies the concepts of the swarm algorithm and learning algorithm. Its process consists of ()3 sub-processes of evaluation, selection and learning. Evaluation involves checking that the students in the class have a score corresponding to the criteria. If a student's score breaks the criteria, then that student is sorted out of the class, and a new student is established. Selection involves choosing the students for passing the class. A student who passes the class is represented by the optimal value and becomes the basis for new student establishment and training of the students who remain in the class. Learning involves training of the students in the class to achieve the criteria of passing the class. Learning classifies the students into 2 groups: the good score group and the bad score group. The good score group is trained based on self-knowledge and students who pass the class, while the bad score group is trained based on the good score group.

The process of establishing new students is defined as follows:

$$S(K)_{new} = \frac{W(K)\sum_{n=1}^{N} f_i(K)s_i(K)}{N} \tag{9}$$

where each student is represented by $S(K)$; K_P, K_I and K_D are represented by K; the score for each student is represented by s_i; the number of students in the class is represented by N; the weight of training for each student is represented by $W(K)$; and the number of scores s_i in the class is represented by $f_i(K)$.

The definition of the learning process for the good score group is given in Equation 10.

$$S(K) = \frac{W(K)\sum_{n=1}^{N} s_i(K)}{N} \tag{10}$$

For the bad score group, the learning process is defined as in Equation 11.

$$S(K) = \frac{W(K)(S_{Bad}(K) + S_{Good}(K))}{2} \tag{11}$$

where S_{Good} is the best score of the good score group and S_{Bad} is the score of a student who is in the bad score group and needs to learn.

The flow of the SLP algorithm, as shown in Figure 2, starts with initial student randomization, in which each student is represented by K_P, K_I and K_D, and then, the score for each student corresponding to the objective function is calculated. Next, each student is evaluated by the score to determine which students should remain in the class. If a student is sorted out, then a new student is established. Finally, each student is classified into 2 groups and trained until they can pass the criteria of the class. In the learning process, students are trained based on the Q-learning algorithm.

B. DETERMINISTIC Q-SLP ALGORITHM WITH A STABLE LEARNING RATE

The Q-learning algorithm is an offline rule of reinforcement learning (RL) [48], [49]. It approximates and updates the current rule with the optimal action-value (Q_*) based on the action-value (Q). This paper proposes the rule for updating given in Equation 13. The pseudocode of the Q-learning algorithm is presented as follows:

Initialization:
1: initial $Q(x, u)$
LOOP Process
2: **while** (episode does not end) **do**
3: Initialize state S_i

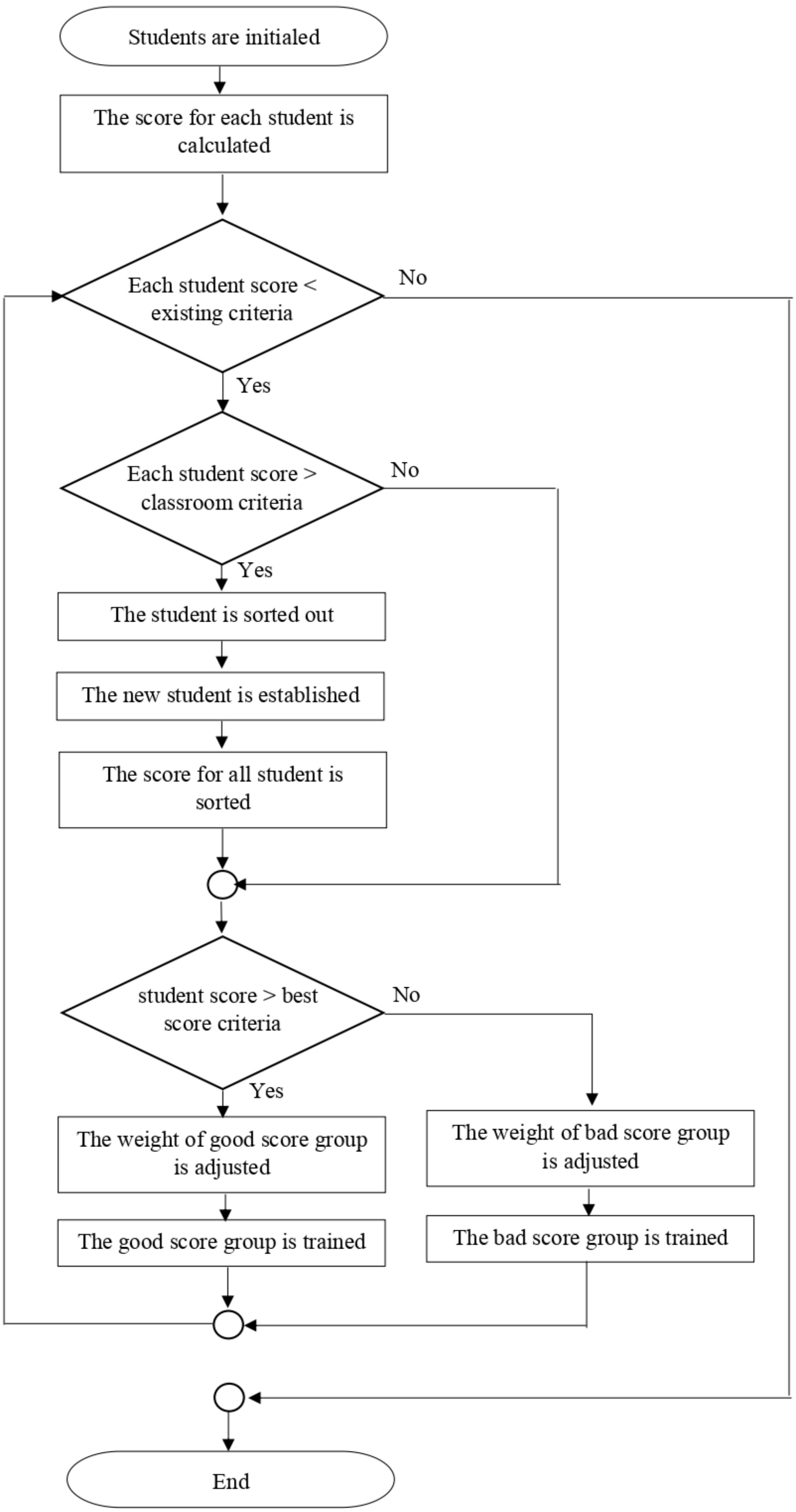

FIGURE 2. **Flow chart of the SLP algorithm.**

while (s_i does not reach the terminal state s_I) **do**
  Perform action u_i, and observe x_{i+1} and R_{i+1}
  Update Q values as in Equation 13.
  $u_i \leftarrow u_{i+1}$
 end while
end while

In the pseudocode, x is the observed state, and u is the action. R is the reward for each episode i.

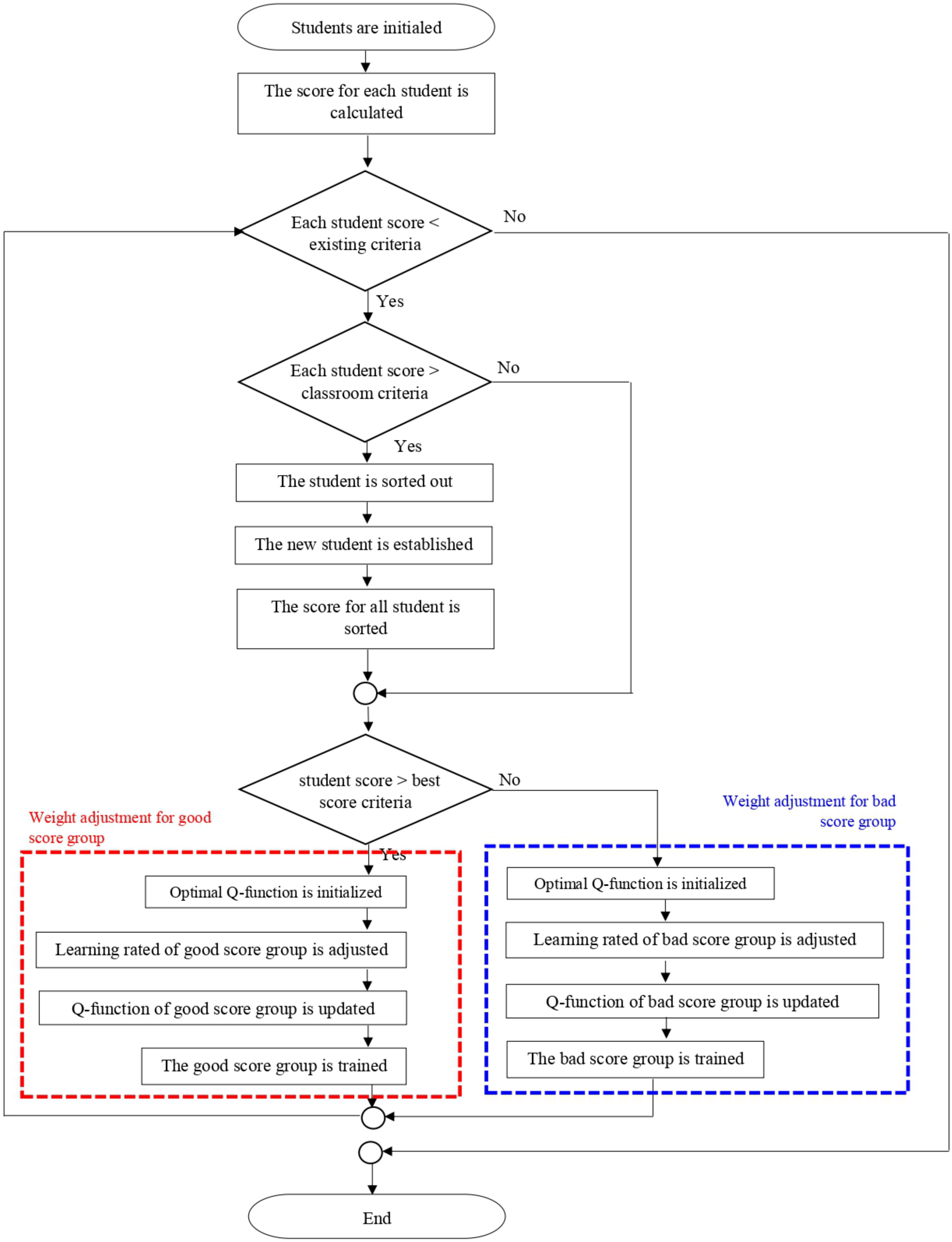

FIGURE 3. Flow chart of the deterministic Q-SLP algorithm.

The deterministic Q-SLP algorithm is a combination of the Q-learning algorithm and SLP algorithm. The Q-learning algorithm is used to adjust the weight of the SLP algorithm as $W(t) = Q_i(x(k), u(k))$, where $Q_i(x(k), u(k))$ is the quality of the Q-function for which the basic equation used is the Bellman equation as follows: [49]

$$Q_i(x(k), u(k)) = J(x(k), u(k)) + \lambda min(Q_i(x(k+1), u(k+1))) \tag{12}$$

where $J(x(k), u(k))$ is the reward received, which, in this paper, is the value of the cost function, and λ is the learning rate.

The updating weight of the SLP algorithm is $W(t+1) = Q_{i+1}(x(k), u(k))$, where $Q_{i+1}(x(k), u(k))$ is the iterative updating of the Q-function at arbitrary state $x(k)$ and control $u(k)$, for which this paper proposes a new deterministic optimal Q-function to define it. The deterministic optimal Q-function is defined as follows:

$$Q_{i+1}(x(k), u(k)) = J(x(k), u(k)) + \lambda_i((1-\mu)Q_i(x(k), u(k)) + \alpha min Q_i(x(k+1), u(k+1))) \tag{13}$$

where λ_i is the learning rate of the Q-function generated based on the closed-loop stabilization of the learning rate of the deterministic Q-learning algorithm as in Equation 15. μ is the factor of the Q-function at iteration i. Note that the deterministic Q-SLP algorithm provides a quantitative methodology for selecting the controller parameters to approach an optimal transient response where the relative stability can be considered by the transient response in each time of tuning.

In the initialization, the proposed method is initiated as

$$Q_0(x(k), u(k)) = \Upsilon(x(k), u(k)) \tag{14}$$

where $\Upsilon(x(k), u(k))$ is a positive semi-definite function.

Theorem 1: At sampling time t, the system with PID control that is tuned according to K_P, K_I and K_D by the deterministic Q-SLP algorithm is stable if the learning rate of the deterministic Q-function is generated as follows:

$$3\nu(t)\left(\frac{\rho(t)-\psi(t)+\beta(t)}{2t\xi(t)+6\nu(t)\beta(t)}\right) < \lambda \leq 3\nu(t)\left(\frac{\rho(t)+\psi(t)+\beta(t)}{2t\xi(t)+6\nu(t)\beta(t)}\right) \tag{15}$$

where

$$\rho(t) = \sqrt{\frac{t\varpi(t)}{\upsilon(t)} + 4NX(t)\lambda(t)},$$

$$\nu(t) = \frac{t(W(t)W(t-1)E_{pp}^2(t))^2}{\upsilon(t)}$$

and $\beta(t) = NW(t)X(t)$.

Proof: The stability of the learning rate is proven according to the Lyapunov function, which can be defined as follows:

$$V(t) = \varphi\int_0^t tE^2(t)dt + (1-e^{-\beta})(m_p(t)+E(t)) + e^{-\beta}(t_s(t)-t_r(t)) \tag{16}$$

The Lyapunov function can be changed as follows:

$$\begin{aligned}\Delta V(t) &= V(t+1) - V(t)\\ &= \varphi\int_0^t tE^2(t+1)dt - \varphi\int_0^t tE^2(t)dt + ((1-e^{-\beta})\\ &\quad \times(m_p(t+1)+E(t+1)) + e^{-\beta}(t_s(t+1)-t_r(t+1))\\ &\quad -(1-e^{-\beta})(m_p(t)+E(t)) + e^{-\beta}(t_s(t)-t_r(t)))\end{aligned} \tag{17}$$

$$= \varphi\int_0^t t\Delta E^2(t)dt + (1-e^{-\beta})(\Delta m_p(t) + \Delta E(t)) + e^{-\beta}(\Delta t_s(t) - \Delta t_r(t)) \tag{18}$$

From the structure of the SLP algorithm as shown in Equation 10 and Equation 11,

$$\Delta E(t) = \Delta W(t)E_p(t) = -\frac{(\lambda(t)W(t) - W(t-1))E_p(t)}{N} \tag{19}$$

where $E_p(t) = \sum_{i=1}^{M}\frac{\partial E_i(t)}{\partial W(t)}$, and M is the number of students.

$$\Delta m_p(t) = \Delta W(t)M_{pp}(t) = -\frac{(\lambda(t)W(t) - W(t-1))M_{pp}(t)}{N} \tag{20}$$

where $M_{pp}(t) = \sum_{i=1}^{M}\frac{\partial m_pi(t)}{\partial W(t)}$.

$$\Delta t_s(t) = \Delta W(t)T_{sp}(t) = -\frac{(\lambda(t)W(t) - W(t-1))T_{sp}(t)}{N} \tag{21}$$

where $T_{sp}(t) = \sum_{i=1}^{M}\frac{\partial T_si(t)}{\partial W(t)}$.

$$\Delta t_s(t) = \Delta W(t)T_{rp}(t) = -\frac{(\lambda(t)W(t) - W(t-1))T_{rp}(t)}{N} \tag{22}$$

where $T_{rp}(t) = \sum_{i=1}^{M}\frac{\partial t_ri(t)}{\partial W(t)}$.
Therefore, Equation 11 can be written as follows:

$$\begin{aligned}\Delta V(t) &= \varphi\int_0^t t\left(\frac{(-\lambda(t)W(t) - W(t-1)E_p(t))^2}{N}\right)dt\\ &\quad -\frac{\lambda(t)W(t) - W(t-1)}{N}((1-e^{-\beta})(M_{pp}(t) - E_P(t))\\ &\quad + e^{-\beta}(T_{sp}(t) - T_{rp}(t)))\end{aligned} \tag{23}$$

If at sampling time t, $\beta_I(t) > \beta_M(t) < \beta_A(t)$, where $\beta_I(t) = t(\frac{(-\lambda(t)W(t)-W(t-1)E_p(t))^2}{N})$, $\beta_M(t) = \frac{\lambda(t)W(t)-W(t-1)}{N}((1 - e^{-\beta})(M_{pp}(t) - E_P(t))$ and $\beta_A(t) = e^{-\beta}(T_{sp}(t) - T_{rp}(t)))$, is satisfied, then the stability based on Lyapunov stability is guaranteed. This means that the learning of the deterministic Q-SLP algorithm is stable when the learning rate (λ) satisfies Equation 15. ■

IV. CONVERGENCE ANALYSIS OF THE DETERMINISTIC Q-FUNCTION

Definition 1 ([40]): The convergence of the Q-function can be proven by generating the sequence of the Q-function ($Q_i(x, u)$), where $i \in \mathbf{Z}^+ + \{0\}$; then,

1. $Q_*(x(k), u(k)) \le Q_{i+1}(x(k), u(k)) \le Q_i(x(k), u(k))$ and
2. $\lim_{i\to\infty} Q_i(x(k), u(k)) = Q^*(x(k), u(k))$.

Lemma 1: For iteration $i \in \mathbf{Z}^+ + \{0\}$, if $Q_i(x(k), u(k))$ can be updated with the learning rate corresponding to Equation 15 and $Q_0(x(k), u(k))$ can be defined as Equation 14, then $Q_*(x(k), u(k)) \le Q_{i+1}(x(k), u(k)) \le Q_i(x(k), u(k))$.

Proof: At sequence $i = 0$, $Q_0(x(k), u(k))$ is initiated with the positive semi-definite function given in Equation 14. Therefore, the above holds for the case of $Q_0(x(k), u(k)) = Q_0(x(k), u(k))$.

At sequence $i + 1$, $\lambda_{i+1} = 1$, and $\mu = 0$.

$$\begin{aligned} &Q_{i+1}(x(k), u(k)) \\ &\quad = J(x(k), u(k)) + \lambda_{i+1}((1-\mu)Q_{i+1}(x(k), u(k)) \\ &\qquad +\alpha min_{u(k+1)}\{Q_{i+1}(x(k+1), u(k+1))\} \\ &\quad = Q_i(x(k), u(k)) + \lambda_0((1-\mu)Q_0(x(k), u(k)) \\ &\quad \le Q_i(x(k), u(k)) \end{aligned} \tag{24}$$

At sequence i, $\lambda_{i+1} = 1$, and $\mu = 0$.

$$\begin{aligned} Q_i(x(k), u(k)) &= J(x(k), u(k)) + \alpha min_{u(k+1)}\{Q_{i+1}(x(k+1), \\ &\qquad \times u(k+1))\} \\ &\ge J(x(k), u(k)) + \alpha min_{u(k+1)}\{Q_*(x(k+1), \\ &\qquad \times u(k+1))\} \\ &= Q_*(x(k), u(k)) \end{aligned} \tag{25}$$

From Equations 24 and 25, we can conclude that $Q_*(x(k), u(k)) \le Q_{i+1}(x(k), u(k)) \le Q_i(x(k), u(k))$. ∎

Lemma 2: Let $\zeta(x(k), u(k)) = J(x(k), u(k)) + \lambda min_{k\to\infty} (x(k+1), u(k+1)) - \mu Q_\infty(x(k), u(k))$, where $x(k), u(k) \ne 0$. Let ϕ be a small positive number; if $\zeta(x(k), u(k)) > 0$, then $\zeta(x(k), u(k)) - \phi > 0$, and if $\zeta(x(k), u(k)) < 0$, then $\zeta(x(k), u(k)) + \phi < 0$; then, $Q_\infty(x(k), u(k))$ is finite.

Proof: From Equation 13, in the case of $\lim_{i\to\inf} \lambda_i \ne 0$,

$$\begin{aligned} Q_\infty(x(k), u(k)) &= J(x(k), u(k)) + (1-\mu)Q_\infty(x(k), u(k)) \\ &\quad +\alpha min_{u_k\to\infty}(x(k+1), u(k+1)) \end{aligned} \tag{26}$$

Assumption 1: From Equation 26, it is assumed that if $\lim_{i\to\infty} \lambda_i = 0$ is true, then $\lim_{i\to\infty} Q_i(x(k), u(k))$ exist.

Given this,

$$\begin{aligned} \zeta(x(k), u(k)) = J(x(k), u(k)) &+ \alpha min_{u(k)\to\infty}(x(k+1), u(k+1)) \\ &- \mu Q_\infty(x(k), u(k)) \end{aligned} \tag{27}$$

where $\zeta(x(k), u(k)) \ne 0$ and $x(k), u(k) \ne 0$.

$\forall \phi > 0$ and $N > 0$, where $N \in \mathbf{R}^+ - \{0\}$, if $\zeta(x(k), u(k)) + \phi$ and $\zeta(x(k), u(k)) - \phi$ have the same sign, then

$$\begin{aligned} \zeta(x(k), u(k)) - \phi &\le J(x(k), u(k)) + \alpha Q min_{u(k+1)}(x(k+1), \\ &\qquad u(k+1)) - \mu Q(x(k), u(k)) \\ &= \zeta(x(k), u(k)) + \phi \end{aligned} \tag{28}$$

On the other hand, from Equation 13,

$$\begin{aligned} Q_{i+1}(x(k), u(k)) &= J(x(k), u(k)) + \lambda_i((1-\mu)Q_i(x(k), u(k))) \\ &\quad +\alpha min_{u(k+1)}Q_i(x(k+1), u(k+1)) \\ Q_i(x(k), u(k)) &= J(x(k), u(k)) + \lambda_{i-1}((1-\mu)Q_{i-1}(x(k), u(k))) \\ &\quad +\alpha min_{u(k+1)}Q_{i-1}(x(k+1), u(k+1)) \\ &\ \ \vdots \\ Q_1(x(k), u(k)) &= J(x(k), u(k)) + \lambda_0((1-\mu)Q_0(x(k), u(k))) \\ &\quad +\alpha min_{u(k+1)}Q_0(x(k+1), u(k+1)) \end{aligned} \tag{29}$$

Therefore, it can be obtained that

$$\begin{aligned} &Q_{i+1}(x(k), u(k)) \\ &\quad = J(x(k), u(k)) + \sum_{j=0}^{i} \lambda_j(((1-\mu)Q_j(x(k), u(k))) \\ &\qquad +\alpha min_{u(k+1)}Q_j(x(k+1), u(k+1))) \end{aligned} \tag{30}$$

Let the limit of i approach ∞. For $x(k)$ and $u(k)$,

$$\begin{aligned} &Q_\infty(x(k), u(k)) \\ &\quad = J(x(k), u(k)) + \sum_{j=0}^{N-1} \lambda_j(((1-\mu)Q_j(x(k), u(k))) \\ &\qquad +\alpha min_{u(k+1)}Q_j(x(k+1), u(k+1))) \\ &\qquad + \sum_{i=N}^{\infty} \lambda_i(((1-\mu)Q_i(x(k), u(k))) + \alpha min_{u(k+1)} \\ &\qquad \times Q_i(x(k+1), u(k+1))) \end{aligned} \tag{31}$$

From Equation 28, $J(x(k), u(k)) + \sum_{j=0}^{N-1} \lambda_j(((1-\mu)Q_j(x(k+1), u(k+1))) + \alpha min_{u(k+1)}Q_j(x(k+1), u(k+1))) + (\zeta(x(k), u(k)) - \phi) \sum_{i=N}^{\infty} \lambda_i(Q_i(x(k), u(k))) \le Q_\infty(x(k), u(k)) < J(x(k), u(k)) + \sum_{j=0}^{N-1} \lambda_j(((1-\mu)Q_j(x(k+1), u(k+1))) + \alpha min_{u(k+1)}Q_j(x(k+1), u(k+1))) + (\zeta(x(k), u(k)) + \phi) \sum_{i=N}^{\infty} \lambda_i(Q_i(x(k), u(k)))$. For $\sum_{j=N}^{\infty} \lambda_j \to \infty$, $Q_\infty(x(k), u(k)) > \infty$; if $\zeta(x(k), u(k)) > 0$, then $\zeta(x(k), u(k)) - \phi > 0$, and if $\zeta(x(k), u(k)) < 0$, then $\zeta(x(k), u(k)) + \phi < 0$. This means that $Q_\infty(x(k), u(k))$ is finite. Therefore, Assumption 1 is true, and the conclusion holds. ∎

Theorem 2: Let $Q_i(x(k), u(k))$ be updated as in Equation 13, where $i \in \mathbf{Z}^+ + \{0\}$, the learning rate is generated as in Equation 15 and $0 \le \mu < 1$. $Q_i(x(k), u(k))$ can approach its $Q_*(x(k), u(k))$ of the Bellman equation for all sequences i, where $i = 1 \to \infty$. That is,

$$\lim_{i\to\infty} Q_i(x(k), u(k)) = Q^*(x(k), u(k)) \tag{32}$$

where $Q^*(x(k), u(k)) = min_{u(k)}\{Q(x(k), u(k)) + \alpha Q_*(x(k+1), u(k+1))\}$.

Proof: From Lemma 2 and Equation 26, let $N > 0$, where $N \in \mathbf{R}^+$; then,

$$\begin{aligned} &Q_\infty(x(k), u(k)) \\ &\quad = J(x(k), u(k)) + (1-\mu)Q_\infty(x(k), u(k)) \\ &\qquad +\alpha min_{u_k\to\infty(x(k+1), u(k+1))} \\ &minQ_\infty(x(k), u(k)) \\ &\quad = min_{u_k}(J(x(k), u(k) + (1-\mu)(J(x(k), \\ &\qquad u(k)) + \ldots + (1-\mu)(J(x(k+N), \\ &\qquad u(k+N)) + \alpha min_{u(x\to k)}(J(x(k+1), \end{aligned}$$

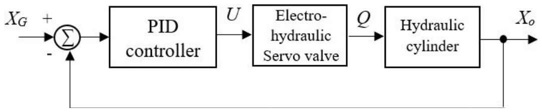

FIGURE 4. The structure diagram of the CPC system with a PID controller.

$$u(k+1)) + (1-\mu)(J(x(k+N-1), u(k+N-1)) + \alpha min_{u(k\to N)} Q_\infty(x(k+N), u(k+N)))))))) \quad (33)$$

With respect to the Bellman principle of optimization [50], Equation 33 can be applied as follows:

$$\lim_{N\to\infty} (1-\mu)Q_\infty(x(k+N), u(k+N)) = 0 \quad (34)$$

and

$$\lim_{N\to\infty} \alpha min Q_\infty(x(k+N), u(k+N)) = 0 \quad (35)$$

From Definition 1 of the optimal Q-function, it can be obtained that

$$Q_*(x(k), u_*(x(k))) \le min_{u(k+N)} Q_\infty(x(k+N), u(k+N)) \quad (36)$$

According to Definition 1 and Equation 26,

$$\begin{aligned} Q_\infty(x(k), u(x(k))) &= J(x(k), u(k)) + (1-\mu)Q_\infty(x(k), u(k)) \\ &\quad + \alpha min_{u(k\to\infty)} Q_\infty(x(k+1), u(k+1)) \\ &= R(x(k), u(k)) + \alpha min_{u(k\to\infty)} Q_*(x(k+1), \\ &\quad u(k+1)) \\ &= Q_*(x(k), u(k)) \end{aligned} \quad (37)$$

This means that $\lim_{i\to\infty} Q_i(x(k), u(k)) = Q^*(x(k), u(k))$ for all sequences i. Therefore, Equation 15 can be proven as optimal based on the Bellman equation. With respect to the conclusion of Lemma 1 and Theorem 2, the convergence of the new deterministic Q-function given in Equation 13 can be proved according to Definition 1. ■

V. SIMULATION AND DISCUSSION

A. MODEL OF CENTRAL POSITION CONTROL

To verify the performance and convergence in autotuning the PID parameter, this paper uses a CPC system as a platform for the verification.

The CPC system is the most important part of a strip steel manufacturing line. It improves the yield of the strip to prevent transverse deviation of the strip. An electro-hydraulic servo valve is used in the CPC system. The structure diagram of the CPC system with a PID controller is shown in Figure 4.

From Figure 4, the main system model is composed of the electro-hydraulic servo valve, hydraulic cylinder and displacement detection sensor. The transfer function of the electro-hydraulic servo valve can be written as follows [45]:

$$G_{sv}(s) = \frac{Q(s)}{U(s)} = \frac{K_{sv}}{\frac{s^2}{\omega_{sv}^2} + \frac{2\zeta_{sv}}{\omega_{sv}}s + 1} \quad (38)$$

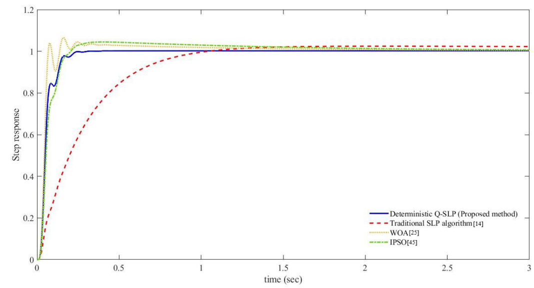

FIGURE 5. Comparison of transient responses in the case of $\beta = 0.5$.

where K_{sv} is the gain of the servo valve with no load, which is set as 0.00196 $m^3/(sA)$. ζ_{sv} is the damping ratio of the servo valve, which is set as 0.7, and ω_{sv} is the natural frequency of the servo valve, which is set as 157 rad/sec. The transfer function of the hydraulic cylinder obtained by neglecting the load uncertainty and quality can be written as follows:

$$G_h(s) = \frac{X_o(s)}{Q(s)} = \frac{\frac{1}{A_p}}{s(\frac{s^2}{\omega_h^2} + \frac{2\zeta_h}{\omega_h}s + 1)} \quad (39)$$

where X_o is the displacement of the hydraulic cylinder. ω_h is the natural frequency of the hydraulic cylinder, which is set as 88 rad/sec. ζ_h is the damping ratio of the hydraulic cylinder, which is set as 0.3. Actually, the system is a combination of complexity and many disturbances. Thus, for without loss of generality and for easy analysis of the transient response in each time of autotuning [45], the mathematical model of the electro-hydraulic servo valve system can be just simplified by neglecting the external noise and the load.

B. SIMULATION RESULT

In this paper, the performance and convergence in autotuning the PID parameter are verified by comparing the simulation results of the deterministic Q-SLP algorithm with those of the traditional SLP algorithm [14], the WOA [25] and the IPSO [45] based on the CPC system when changing the constant of the objective function (β). The changing constant of the objective function given in Equation 8 is set to $\beta = 0.5$, $\beta = 1$ and $\beta = 1.5$. In the simulation, the number of nodes for each algorithm is 20, with the same initial value, simulation time of 5 sec, and ranges of K_P, K_I and K_D of [0,20], [0,20] and [0,1]. The maximum number of iterations is 150 iterations.

The comparison results of the performance and convergence in the case of $\beta = 0.5$ are shown as Figures 5 and 6. In the case of $\beta = 1$, the comparison results are shown in Figures 7 and 8. Figures 9 and 10 show the comparison results of the performance and convergence in the case of $\beta = 1.5$.

The comparative performance indices, such as M_p, t_s, t_r and e_{ss}, based on CPC system for $\beta = 0.5$ are shown in Figure 5, and Table 1. The comparison of the proposed method yields $t_s = 0.113$ sec, $t_r = 0.107$ sec, $M_p = 0.002$ mm and $e_{ss} = 0.002$ mm. The traditional SLP algorithm [14] yields $t_s = 2.006$ sec, $t_r = 0.220$ sec, M_p 0.025 mm and

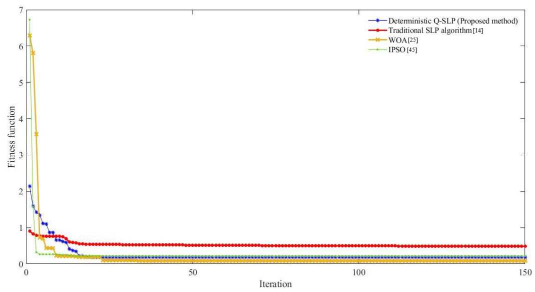

FIGURE 6. Comparison of objective functions in the case of $\beta = 0.5$.

TABLE 1. Summary of the performance and convergence comparison for $\beta = 0.5$.

Algorithm	Q-SLP	Traditional SLP [14]	WOA [25]	IPSO [45]
K_P	18.226	3.938	23.667	15.001
K_I	0.523	0.395	19.326	10.858
K_D	0.011	0.033	0.011	0.001
t_s (sec)	0.113	2.006	0.120	0.449
t_r (sec)	0.107	0.220	0.110	0.127
M_p (mm)	0.002	0.025	0.066	0.045
e_{ss} (mm)	0.002	0.023	0.003	0.006
$J(t)$	0.0052	0.517	0.082	0.216

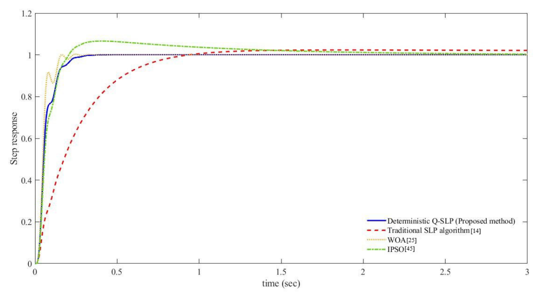

FIGURE 7. Comparison of transient responses in the case of $\beta = 1$.

$e_{ss} = 0.023$ mm. The WOA [25] yields $t_s = 0.120$ sec, $t_r = 0.110$ sec, $M_p = 0.066$ mm and $e_{ss} = 0.003$ mm. Finally, the IPSO [45] yields $t_s = 0.449$ sec, $t_r = 0.127$ sec, $M_p = 0.045$ mm and $e_{ss} = 0.006$ mm. The comparative convergence curve between the proposed method, the traditional SLP algorithm [14], the WOA [25] and the IPSO [45] is shown in Figure 6 and Table 1. The minimizing result of the fitness function for the proposed method is 0.0052 and takes 20 iterations; for the traditional SLP algorithm [14] is 0.517 and takes 50 iterations; for the WOA [25] is 0.082 and takes 54 iterations; for the IPSO [45] is 0.216 and takes 45 iterations.

The comparative performance indices, such as M_p, t_s, t_r and e_{ss} based on CPC system for $\beta = 1$ are shown in Figure 7, and Table 2. The comparison of the proposed method yields $t_s = 0.403$ sec, $t_r = 0.102$ sec, $M_p = 0.001$ mm and $e_{ss} = 0.001$ mm. The traditional SLP algorithm [14] yields $t_s = 1.789$ sec, $t_r = 0.189$ sec, $M_p = 0.023$ mm and $e_{ss} = 0.021$ mm. The WOA [25] yields $t_s = 0.417$ sec, $t_r = 0.117$ sec, $M_p = 0.003$ mm and $e_{ss} = 0.001$ mm.

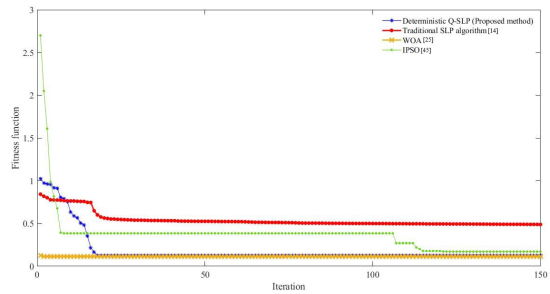

FIGURE 8. Comparison of objective functions in the case of $\beta = 1$.

TABLE 2. Summary of the performance and convergence comparison for $\beta = 1$.

Algorithm	Q-SLP	Traditional SLP [14]	WOA [25]	IPSO [45]
K_P	15.904	4.426	20.276	13.492
K_I	0.314	0.478	10.323	14.013
K_D	0.022	0.041	0.002	0.013
t_s (sec)	0.403	1.789	0.417	0.473
t_r (sec)	0.102	0.189	0.117	0.130
M_p (mm)	0.001	0.023	0.003	0.066
e_{ss} (mm)	0.001	0.021	0.001	0.004
$J(t)$	0.112	0.499	0.113	0.171

Finally, the IPSO [45] yields $t_s = 0.473$ sec, $t_r = 0.130$ sec, $M_p = 0.066$ mm and $e_{ss} = 0.004$ mm. The comparative convergence curve between the proposed method, the traditional SLP algorithm [14], the WOA [25] and the IPSO [45] is shown in Figure 8 and Table 2. The minimizing result of the fitness function for the proposed method is 0.112 and takes 18 iterations; for the traditional SLP algorithm [14] is 0.499 and takes 96 iterations; for the WOA [25] is 0.113 and takes 22 iterations; for the IPSO [45] is 0.171 and takes 115 iterations.

The comparative performance indices, such as M_p, t_s, t_r and e_{ss} based on CPC system for $\beta = 1.5$ are shown in Figure 9, and Table 3. The comparison of the proposed method yields $t_s = 0.117$ sec, $t_r = 0.100$ sec, $M_p = 0.002$ mm and $e_{ss} = 0.002$ mm. The traditional SLP algorithm [14] yields $t_s = 2.006$ sec, $t_r = 0.220$ sec, $M_p = 0.025$ mm and $e_{ss} = 0.023$ mm. The WOA [25] yields $t_s = 0.120$ sec, $t_r = 0.110$ sec, $M_p = 0.066$ mm and $e_{ss} = 0.003$ mm. Finally, the IPSO [45] yields $t_s = 0.450$ sec, $t_r = 0.127$ sec, $M_p = 0.045$ mm and $e_{ss} = 0.0064$ mm. The comparative convergence curve between the proposed method, the traditional SLP algorithm [14], the WOA [25] and the IPSO [45] is shown in Figure 10 and Table 3. The minimizing result of the fitness function for the proposed method is 0.0069 and takes 14 iterations; for the traditional SLP algorithm [14] is 0.209 and takes 35 iterations; for the WOA [25] is 0.082 and takes 16 iterations; for the IPSO [45] is 0.216 and takes 98 iterations.

From the comparative results, it can be clearly observed that the proposed method can provide 4 performance indices and convergence superior to those of the traditional SLP algorithm [14], the WOA [25] and IPSO [45].

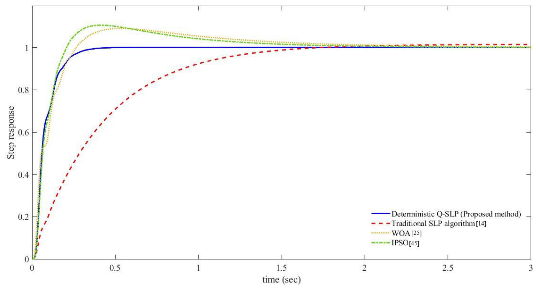

FIGURE 9. Comparison of transient responses in the case of $\beta = 1.5$.

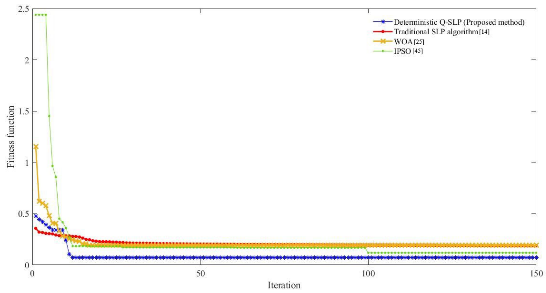

FIGURE 10. Comparison of objective functions in the case of $\beta = 1.5$.

TABLE 3. Summary of the performance and convergence comparison for $\beta = 1.5$.

Algorithm	Q-SLP	Traditional SLP [14]	WOA [25]	IPSO [45]
K_P	18.226	3.934	23.666	15.000
K_I	0.523	0.395	19.326	10.858
K_D	0.011	0.032	0.011	0.001
t_s (sec)	0.117	2.006	0.120	0.450
t_r (sec)	0.100	0.220	0.110	0.127
M_p (mm)	0.002	0.025	0.066	0.045
e_{ss} (mm)	0.002	0.023	0.003	0.0064
$J(t)$	0.0069	0.209	0.082	0.216

Remark 1: According to the comparative performance indices based on the transient response analysis and the convergence result for 3 cases study, the proposed method is lowest the value of performance indices, objective function and iteration of tuning since in every time of autotuning, the proposed method records the current of the PID parameter and the current value of objective function to the database and uses it to determine the new PID parameter. Furthermore, in the PID parameter autotuning processing, the proposed method adjusts the PID parameters by considering the learning rate with the current transient response as Equation 15. Note that the traditional SLP algorithm [14] autotunes the PID parameters under the random processing while the WOA [25] and the IPSO [45] autotune the PID parameters under the neighbor value and the initial value. Therefore, it can be concluded that the PID controller which is auto tuned by using deterministic Q-SLP algorithm with fitness function as Equation 8 based on CPC control system has a high performance, more stable and faster tuning than the other algorithm.

VI. CONCLUSION

In this paper, a deterministic Q-SLP algorithm is proposed to optimize and improve the stability, convergence and performance in autotuning the PID parameter. It differs from the traditional SLP algorithm [14], and the scheme of adjusting the weight of the SLP algorithm is presented based on the new deterministic Q-learning algorithm. The global optimization of the proposed method is proven based on the Bellman equation. Additionally, this paper proposes closed-loop stabilization of the learning rate to improve the stability of the learning process of the Q-learning algorithm, for which the stability is proven according to the Lyapunov stability theorem. To confirm the performance and convergence, the simulation results of autotuning the PID controller for the CPC system are compared. The comparison shows that the proposed method can produce better results than the traditional SLP algorithm, the WOA [24] and the IPSO [45]. According to the simulation results, it can be concluded that the theoretical approach in this paper achieves the performance indices, i.e, t_s, t_r, M_p and e_{ss}, and convergence of the optimal autotuning of the PID parameter suitable for practical applications. However, the time-delay and the disturbance can be easily found in many real physical control problems which may make the system convergence and the system performance out of control. Therefore, the new deterministic Q-SLP algorithm for an uncertain nonlinear system with time-varying delay can be investigated in future research work.

REFERENCES

[1] C. Chen and J. Lu, "Design of self-tuning SISO partial-form model-free adaptive controller for vapor-compression refrigeration system," *IEEE Access*, vol. 7, pp. 125771–125782, 2019.

[2] J. L. Meza, V. Santibanez, R. Soto, and M. A. Llama, "Fuzzy self-tuning PID semiglobal regulator for robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 59, no. 6, pp. 2709–2717, Jun. 2012.

[3] Z. Chen, Y. Yuan, X. Yuan, Y. Huang, X. Li, and W. Li, "Application of multi-objective controller to optimal tuning of PID gains for a hydraulic turbine regulating system using adaptive grid particle swam optimization," *ISA Trans.*, vol. 56, pp. 173–187, May 2015.

[4] M. Cetin and S. Iplikci, "A novel auto-tuning PID control mechanism for nonlinear systems," *ISA Trans.*, vol. 58, pp. 292–308, Sep. 2015.

[5] E. O. Freire, F. G. Rossomando, and C. M. Soria, "Self-tuning of a neuro-adaptive PID controller for a SCARA robot based on neural network," *IEEE Latin Amer. Trans.*, vol. 16, no. 5, pp. 1364–1374, May 2018.

[6] L. R. R. dos Santos, F. R. Durand, and T. Abrao, "Adaptive PID scheme for OCDMA next generation PON based on heuristic swarm optimization," *IEEE Syst. J.*, vol. 13, no. 1, pp. 500–510, Mar. 2019.

[7] L. R. R. dos Santos, F. R. Durand, A. Goedtel, and T. Abrão, "Auto-tuning PID distributed power control for next-generation passive optical networks," *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D110–D125, Oct. 2018.

[8] R. K. Sahu, S. Panda, and P. C. Pradhan, "Design and analysis of hybrid firefly algorithm-pattern search based fuzzy PID controller for LFC of multi area power systems," *Int. J. Elect. Power Energy Syst.*, vol. 69, pp. 200–212, Jul. 2015.

[9] M. N. Anwar and S. Pan, "A new PID load frequency controller design method in frequency domain through direct synthesis approach," *Int. J. Elect. Power Energy Syst.*, vol. 67, pp. 560–569, May 2015.

[10] A. K. Yadav and P. Gaur, "An optimized and improved STF-PID speed control of throttle controlled HEV," *Arabian J. Sci. Eng.*, vol. 41, no. 9, pp. 3749–3760, Sep. 2016.

[11] A. Aldemir, "PID controller tuning based on phase margin (PM) for wireless temperature control," *Wireless Pers. Commun.*, vol. 103, no. 3, pp. 2621–2632, Dec. 2018.

[12] D. Guha, P. K. Roy, and S. Banerjee, "Binary bat algorithm applied to solve MISO-type PID-SSSC-based load frequency control problem," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 43, no. 2, pp. 323–342, Jun. 2019.

[13] M. Rahman, Z. C. Ong, W. T. Chong, S. Julai, and X. W. Ng, "Wind turbine tower modeling and vibration control under different types of loads using ant colony optimized PID controller," *Arabian J. Sci. Eng.*, vol. 44, no. 2, pp. 707–720, Feb. 2019.

[14] J. Pongfai, X. Su, H. Zhang, and W. Assawinchaichote, "A novel optimal PID controller autotuning design based on the SLP algorithm," *Expert Syst.*, pp. 1–15, Nov. 2019.

[15] P. Shi, X. Su, and F. Li, "Dissipativity-based filtering for fuzzy switched systems with stochastic perturbation," *IEEE Trans. Autom. Control*, vol. 61, no. 6, pp. 1694–1699, Jun. 2016.

[16] J. Yu, P. Shi, H. Yu, B. Chen, and C. Lin, "Approximation-based discrete-time adaptive position tracking control for interior permanent magnet synchronous motors," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1363–1371, Jul. 2015.

[17] S.-E.-I. Hasseni, L. Abdou, and H.-E. Glida, "Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms," *Evol. Intell.*, pp. 1–13, Nov. 2019. [Online]. Available: https://link.springer.com/article/10.1007%2Fs12065-019-00312-8

[18] J. Kallannan, A. Baskaran, N. Dey and A. S. Ashour, *Computer Science, Engineering & Technology*, 1st ed. Boca Raton, Florida, USA: CRC Press, 2018, pp. 14–19.

[19] A. Noshadi, J. Shi, W. S. Lee, P. Shi, and A. Kalam, "Optimal PID-type fuzzy logic controller for a multi-input multi-output active magnetic bearing system," *Neural Comput. Appl.*, vol. 27, no. 7, pp. 2031–2046, Oct. 2016.

[20] H. Shen and J. Yan, "Optimal control of rail transportation associated automatic train operation based on fuzzy control algorithm and PID algorithm," *Autom. Control Comput. Sci.*, vol. 51, no. 6, pp. 435–441, Nov. 2017.

[21] H.-T. Ye and Z.-Q. Li, "PID neural network decoupling control based on hybrid particle swarm optimization and differential evolution," *Int. J. Autom. Comput.*, pp. 1–6, Nov. 2015.

[22] C.-M. Lin, M.-C. Li, A.-B. Ting, and M.-H. Lin, "A robust self-learning PID control system design for nonlinear systems using a particle swarm optimization algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 4, pp. 225–234, Dec. 2011.

[23] A. Sungthong and W. Assawinchaichote, "Particle swam optimization based optimal PID parameters for air heater temperature control system," *Procedia Comput. Sci.*, vol. 86, pp. 108–111, Jan. 2016.

[24] T. Spoljaric, C. Lusetic, and V. Simovic, "Optimization of PID controller in AVR system by using ant lion optimizer algorithm," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 1–5.

[25] A. M. Mosaad, M. A. Attia, and A. Y. Abdelaziz, "Whale optimization algorithm to tune PID and PIDA controllers on AVR system," *Ain Shams Eng. J.*, vol. 10, no. 4, pp. 755–767, Dec. 2019.

[26] A. H. Heidari, S. Etedali, and M. R. Javaheri-Tafti, "A hybrid LQR-PID control design for seismic control of buildings equipped with ATMD," *Frontiers Struct. Civil Eng.*, vol. 12, no. 1, pp. 44–57, Mar. 2018.

[27] K. M. Goher and S. O. Fadlallah, "PID, BFO-optimized PID, and PD-FLC control of a two-wheeled machine with two-direction handling mechanism: A comparative study," *Robot. Biomimetics*, vol. 5, no. 1, pp. 1–16, Dec. 2018.

[28] T. Samakwong and W. Assawinchaichote, "PID controller design for electro-hydraulic servo valve system with genetic algorithm," *Procedia Comput. Sci.*, vol. 86, pp. 91–94, Jan. 2016.

[29] K. S. Rajesh and S. S. Dash, "Load frequency control of autonomous power system using adaptive fuzzy based PID controller optimized on improved sine cosine algorithm," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 6, pp. 2361–2373, Jun. 2019.

[30] J. Kaliannan, A. Baskaran, N. Dey, and S. A. Ashour, "Ant colony optimization algorithm based PID controller for LFC of single area power system with non-linearity and boiler dynamics," *World J. Model. Simul.*, vol. 12, no. 1, pp. 3–14, 2016.

[31] Z. Bingul, "A new PID tuning technique using differential evolution for unstable and integrating processes with time delay," in *Proc. Int. Conf. Neural Inf. Process.*, 2004, pp. 254–260.

[32] N. Razmjooy, M. Khalilpour, and M. Ramezani, "A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: Theory and its application in PID designing for AVR system," *J. Control, Autom. Electr. Syst.*, vol. 27, no. 4, pp. 419–440, Aug. 2016.

[33] A. Marin, J. A. Hernandez R., and J. A. Jimenez, "Tuning multivariable optimal PID controller for a continuous stirred tank reactor using an evolutionary algorithm," *IEEE Latin Amer. Trans.*, vol. 16, no. 2, pp. 422–427, Feb. 2018.

[34] M. Dangor, O. A. Dahunsi, J. O. Pedro, and M. M. Ali, "Evolutionary algorithm-based PID controller tuning for nonlinear quarter-car electrohydraulic vehicle suspensions," *Nonlinear Dyn.*, vol. 78, no. 4, pp. 2795–2810, Dec. 2014.

[35] M. Ghanamijaber, "A hybrid fuzzy-PID controller based on gray wolf optimization algorithm in power system," *Evolving Syst.*, vol. 10, no. 2, pp. 273–284, Jun. 2019.

[36] R. K. Mandava and P. R. Vundavilli, "Implementation of modified chaotic invasive weed optimization algorithm for optimizing the PID controller of the biped robot," *Sādhanā*, vol. 43, no. 5, pp. 1–18, May 2018.

[37] R. K. Mandava and R. Pandu Vundavilli, "An optimal PID controller for a biped robot walking on flat terrain using MCIWO algorithms," *Evol. Intell.*, vol. 12, no. 1, pp. 33–48, 2019.

[38] K. Jagatheesan, B. Anand, S. Samanta, N. Dey, V. Santhi, S. A. Ashour, and E. V. Balas, "Application of power pollination algorithm in load frequency control of multi-area interconnected power system with nonlinearity," *Neural Comput. Appl.*, vol. 28, no. 1, pp. 475–488, 2017.

[39] K. Jagatheesan, B. Anand, S. Samanta, N. Dey, A. S. Ashour, and V. E. Balas, "Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 503–515, Mar. 2019.

[40] B. Luo, Y. Yang, and D. Liu, "Adaptive *Q*-learning for data-based optimal output regulation with experience replay," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3337–3348, 2018.

[41] J. Li, T. Chai, L. F. Lewis, Z. Ding, and Y. Jiang, "Off-policy interleaved *Q*-learning: Optimal control for affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1308–1320, Sep. 2019.

[42] M. Guo, Y. Liu, and J. Malec, "A new *Q*-learning algorithm based on the metropolis criterion," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2140–2143, Oct. 2004.

[43] C.-F. Juang, "Combination of online clustering and Q-value based GA for reinforcement fuzzy system design," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 289–302, Jun. 2005.

[44] J. S. Campbell, S. N. Givigi, and H. M. Schwartz, "Multiple model *Q*-learning for stochastic asynchronous rewards," *J. Intell. Robot. Syst.*, vol. 81, nos. 3–4, pp. 407–422, Mar. 2016.

[45] L. Jia and X. Zhao, "An improved particle swarm optimization (PSO) optimized integral separation PID and its application on central position control system," *IEEE Sensors J.*, vol. 19, no. 16, pp. 7064–7071, Aug. 2019.

[46] K. M. Goher and S. O. Fadlallah, "Bacterial foraging-optimized PID control of a two-wheeled machine with a two-directional handling mechanism," *Robot. Biomimetics*, vol. 4, no. 1, pp. 1–19, Dec. 2017.

[47] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 384–391, Jun. 2004.

[48] Q. Shi, H.-K. Lam, B. Xiao, and S.-H. Tsai, "Adaptive PID controller based on *Q*-learning algorithm," *CAAI Trans. Intell. Technol.*, vol. 3, no. 4, pp. 235–244, 2018.

[49] L. F. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, 2009.

[50] Q. Wei, L. F. Lewis, Q. Sun, P. Yan, and R. Song, "Discrete-time deterministic *Q*-learning: A novel convergence analysis," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1224–1237, May 2017.

JIRAPUN PONGFAI received the Bachelor of Engineering degree in computer engineering from Naresuan University, Phitsanulok, Thailand, in 2015, and the Master of Engineering degree in electrical and information engineering from the King Mongkut's University of Technology Thonburi, Bangkok, Thailand, in 2017. Her research interests include intelligence control, optimal and predictive controller design, artificial intelligence, and embedded system design.

XIAOJIE SU received the Ph.D. degree in control theory and control engineering from the Harbin Institute of Technology, China, in 2013. He is currently a Professor and the Associate Dean with the College of Automation, Chongqing University, Chongqing, China. He has published two research monographs and more than 50 research articles in international referred journals. His current research interests include intelligent control systems, advanced control, and unmanned system control. He currently serves as an Associate Editor for a number of journals, including the IEEE Systems Journal, the IEEE Control Systems Letters, *Information Sciences*, and *Signal Processing*. He is also an Associate Editor of the Conference Editorial Board, IEEE Control Systems Society. He was named to the 2017, 2018, and 2019 Highly Cited Researchers list, Clarivate Analytics.

HUIYAN ZHANG received the B.Sc. degree in automation from the Hebei University of Technology, Tianjin, China, in 2012, and the M.Sc. degree in control engineering and the Ph.D. degree in control theory and control engineering from the Harbin Institute of Technology, Harbin, in September 2014 and April 2019, respectively. From September 2015 to September 2017, she was a Joint Training Ph.D. Student with the School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide, Australia. She is currently a Lecturer with Chongqing Technology and Business University. Her research interests include stochastic switched systems, intelligent control systems, model reduction, balanced truncation, robust control, and filtering design.

WUDHICHAI ASSAWINCHAICHOTE received the B.S. degree (Hons.) in electrical engineering from Assumption University, Bangkok, Thailand, in 1994, the M.E. degree in electrical engineering from Pennsylvania State University (Main Campus), PA, USA, in 1997, and the Ph.D. degree in electrical engineering from The University of Auckland, New Zealand, in 2004. He is currently an Associate Professor with the Department of Electronic and Telecommunication Engineering, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand. He has published a research monograph and more than 20 research articles in international refereed journals indexed by SCI/ESCI (Clarivate Analytics). His current research interests include fuzzy control, robust control, optimal control, system and control theory, computational intelligence, and PID controller design. He currently serves as an Associate Editor for a number of journals and also serves as a Reviewer, including *Automatica*; the IEEE Transactions on Industrial Electronics; the IEEE Transactions on Fuzzy Systems; the IEEE Transactions on Cybernetics; the IEEE Transactions on Systems, Man, and Cybernetics: Systems; *Neural Computing and Applications*; and IEEE Access.