IJIE

UTHM Penerbit
Universiti Tun Hussein Onn Malaysia

INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

Vol. 18 No. 1 (2026) 84-98
https://publisher.uthm.edu.my/ojs/index.php/ijie

A Nonlinear PID Control for Time-varying Batch Processes by Using Particle Swarm Optimization

Zhiwen Wang[1]*, Shunfeng Ji[1], Amirul Syafiq Sadun[2], Zheng Chen[1], Qun Zhuang[1]

[1] *School of New Materials and Shoes & Clothing Engineering, Liming Vocational University, Quanzhou 362000, Fujian Province, CHINA*
[2] *Department of Electrical Engineering Technology, Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia (Pagoh Higher Education Hub), Pagoh 84600, Muar, Johor, MALAYSIA*

*Corresponding Author: aiwangzhiwen@163.com
DOI: https://doi.org/10.30880/ijie.2026.18.01.007

Article Info

Received: 12 December 2025
Accepted: 28 March 2025
Available online: 17 April 2026

Keywords

Batch process, nonlinear PID, particle swarm optimization, nonlinear process, time-varing process

Abstract

Nonlinear batch processes in physical and chemical industries often experience time-dependent parameter changes and dynamic perturbations, which make the common PID method insufficient for achieving high-quality control effects. A particle swarm optimization (PSO)–based nonlinear PID (NLPID) control strategy is introduced to handle the nonlinear behavior and time-varying properties commonly observed in batch processes. At the beginning of this study, a nonlinear function is used to replace the tracking error in the PID method, thereby achieving NLPID control. Secondly, considering a performance index of integral square error (ISE), the optimal NLPID control parameters can be obtained by using PSO algorithm (PSO-NLPID). The proposed PSO-NLPID method is ultimately validated in batch process applications through comparative analysis with current methods. To demonstrate the validity of the proposed control strategy, a typical fermentation process is simulated in the batch cycle. Regarding the parameter time-varying problem in nonlinear batch processes, the PSO-NLPID strategy outperforms existing methods, as evidenced by its lowest ISE value of 1.1293, and a performance enhancement exceeding 65% over standard PID control, which indicates that PSO-NLPID has better control performance.

1. Introduction

As an indispensable production method in industrial processes, batch processes are commonly employed for producing high-value products in small quantities and multiple varieties, especially in the manufacturing fields of food processing, biopharmaceuticals, and fine chemicals [1], [2]. Generally, due to the limited batch period of the batch process, process parameters and operating conditions can be more flexibly adjusted according to production requirements to meet customers' personalized needs [3]. In actual industrial processes, the effectiveness of control is an important component of batch production processes in order to complete safely and accurately production tasks as required [4], [5]. However, the batch process itself has strongly nonlinear and highly time-varying characteristics, especially in the transient stages of variable targets and operating ranges [6], which bring many difficulties and challenges to the control of batch processes.

As has been investigated in recent years, existing control technologies have a significant promoting effect on the development of industrial processes [7], [8]. Nowadays, it is almost impossible to find engineering systems that do not use control technology in people's lives. On the one hand, in practical applications, including personalized medical solutions, autonomous driving in automobiles, aerospace engineering, urban traffic management, intelligent manufacturing industry, and other fields, control technology is needed to achieve them; On the other hand, in terms of theoretical research, methods such as data-driven control, adaptive control, PID control, model predictive control, model-free adaptive control, and neural network control are constantly being improved [9]. However, in actual industrial processes, PID control still has a significant impact on over 90% of industries [10]. It is worth emphasizing that although the adoption rate of PID control techniques may decrease, researchers predict that it will still affect nearly 80% of industrial processes in ten years [9], [10]. Therefore, the application research of PID control is still highly concerned.

It can not be denied that PID control possesses notable advantages in practical engineering applications as a data-driven control approach [11]. Since PID control does not require an explicit mathematical model of the plant, it can control linear systems and a few nonlinear systems using only inputs and outputs [12]. In practice, the deviation between the system output and the desired reference is employed to update the control signal, thereby realizing data-driven control [13]. Here, the implementation of ordinary PID control is a linear combination of proportional, integral, and derivative errors [14]. In previous studies, scholars have focused more on optimizing control parameters [15], improving linear structures [16], and combining other strategies [17] in PID control research, ultimately achieving high-performance control results. However, in the essential, these enhanced PID schemes still rely on an unchanged linear aggregation of tracking errors, which cannot effectively address complex nonlinear problems.

Nonlinear control method is an effective mean to solve nonlinear problems, and its control objective is to make the output as close as possible to the set expected target in the controlled process, that is, the tracking error infinitely approaches to zero [18]. In order to adapt to the different working conditions of industrial processes, references [19] and [20] improved the classical PID controller by utilizing the characteristics of nonlinear processes and proposed a nonlinear PID (NLPID) control method to achieve the goal of improving practicality. It is worth noting that the NLPID control can more accurately reflect the nonlinear relationship between the control input and the error signal in the controlled process [21]. In this study, in order to overcome the shortcomings of simple linear PID controllers, the above-mentioned nonlinear PID control method was applied to batch processes to achieve good control effects.

No matter what form of PID control method is developed, the selection of high-quality tuning parameters is critical to achieving superior control performance [22]. Particle swarm optimization (PSO), an intelligent optimization technique inspired by the collective behavior of bird flocks, exhibits stronger global search capability than conventional tuning approaches such as Cohen–Coon, Ziegler–Nichols, and internal model control (IMC) [23]. Previous studies have demonstrated the effectiveness of PSO in optimizing the parameters of linear PID controllers [24]. However, the method of improving NLPID controllers by using PSO algorithm (PSO-NLPID) has seldom been explored in batch process applications. To bridge this gap, this study proposes to apply PSO-NLPID to batch processes to solve their nonlinear and time-varying control problems.

The other parts of the manuscript is organized into distinct sections, beginning with Section **2**, which discusses the preliminary work associated with this research, including problem description of batch processes, basic principles of PSO algorithm, and implementation process of PSO-NLPID method. The Section **3** is simulation research and result discussion, where batch fermentation simulations are employed to validate the proposed control strategy. The Section 4 is about the summary and outlook of this article.

2. Preliminary Works

2.1 Problem Statement

In order to describe the manifestation of state changes in batch processes, this work employs a discrete-time nonlinear single-input single-output (SISO) model, which is formulated by the following functional expression [25], [26]:

$$y(k+1) = f_s\big(y(k), y(k-1),\ldots, y(k-\mathrm{n_a}), u(k), u(k-1),\ldots, u(k-\mathrm{n_b}), v(k), v(k-1),\ldots, v(k-\mathrm{n_c})\big) \tag{1}$$

In which the state function $f_s(\cdot)$ is defined as a nonlinear expression describing the time-varying dynamics of the batch system. Here, k denotes the discrete sampling interval of the controlled system, while N_k represents the total number of sampling instants within the nonlinear process over one batch cycle, $k = 1,2,3,\ldots,N_k$, n_a, n_b, n_c are all positive integers. At the discrete time index k, $y(k) \in \mathbf{R}$ is defined as the system output and $u(k) \in \mathbf{R}$ as the control input, while $v(k) \in \mathbf{R}$ captures disturbance effects, which are generally modeled as additive noise in the output.

System discretization is a mean of solving nonlinear process control problems, which measures the variable signals of the controlled process within a certain sampling period to achieve the goal of implementing control algorithms on digital computers. As the research object of this work, the expected goals of batch processes usually exhibit nonlinear characteristics of dynamic changes in stages. For this reason, the entire batch cycle is completed within a time horizon of T_t, and the sampling period is t_s once every time. It can be seen that a batch cycle can be decomposed into a certain number N_k of moments, and at this time, $N_k = T_t / t_s$ can be obtained. Once the expected value is defined as the tracking goal, the batch process is initialized with the control input and output denoted by $[u_0, y_0]$. The NLPID control [19], [20] structure diagram of the SISO discrete-time nonlinear system described in equation (1) is shown in Fig. 1.

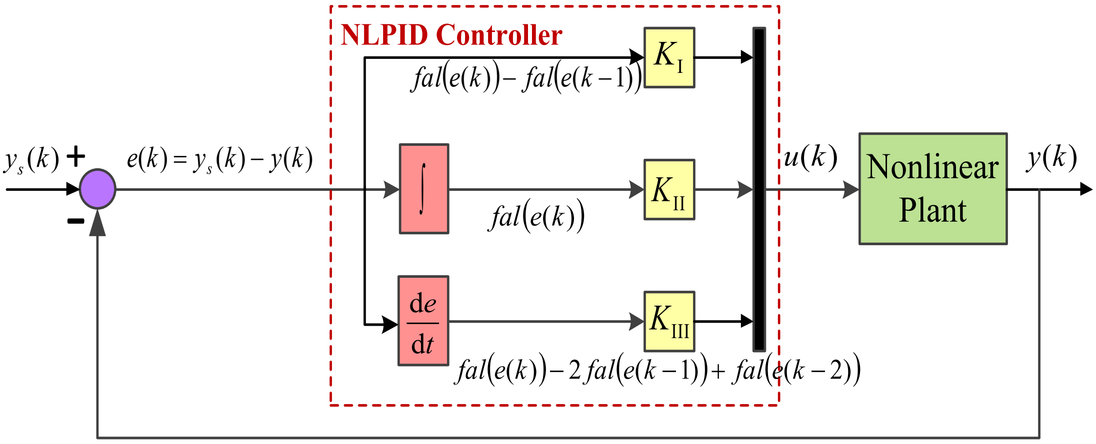

Fig. 1 *Schematic illustration of the NLPID-based control system for a nonlinear batch process*

Fig. 1 illustrates the NLPID control form of a nonlinear system in discrete time. It is not difficult to see that NLPID control can adjust the system input $u(k)$ through certain learning rules at all times to effectively control the system output $y(k)$, ultimately achieving control requirements that continuously approach the desired target. Similarly, there are three control loops related to tracking errors in NLPID control, namely proportional (K_{I}), integral (K_{II}), and derivative (K_{III}), which correspond to three control parameters that need to be tuned: proportional coefficient, integral coefficient, and derivative coefficient. In general, the control law of NLPID in the controlled process, that is, the feedback control formula of the input variable, can be expressed as [19], [20]:

$$u(k) = K_{\mathrm{I}} \cdot \left[fal(e(k)) + \frac{1}{T_i} \cdot \int_0^k fal(e(\tau)) d\tau + T_d \cdot \frac{d}{dk} fal(e(k)) \right] \tag{2}$$

$$e(k) = y_s(k) - y(k) \tag{3}$$

In the formula, $y_s(k)$ represents the expected target of the nonlinear process, which is the given tracking target. The variable $e(k)$ corresponds to the tracking error produced by the control system, while the symbols

K_{I}, T_{i}, and T_{d} are used to indicate the proportional gain coefficient, integral time constant, and derivative time constant, respectively. In this controller design, $fal(\cdot)$ serves as a nonlinear power function that regulates the control input and adjusts the controller gain in response to variations in the input error.

Here, for simplicity, let $fal(e(k)) = fal(e(k), a, \delta)$, and the expression of its function is as follows [19], [20]

$$fal(e(k)) = fal(e(k), a, \delta) = \begin{cases} a \cdot (e(k))^{a-1}, & e(k) > \delta \\ \dfrac{1}{\delta^{1-a}}, & 0 < e(k) \le \delta \end{cases} \tag{4}$$

where, a is a power parameter or smoothness factor, which characterizes the degree of roughness of the nonlinear function $fal(\cdot)$, δ is a threshold parameter or exponential factor that characterizes the linear interval size of a nonlinear function $fal(\cdot)$.

Under a sampling interval of t_{s}, the learning mechanism of the NLPID controller for discrete-time nonlinear batch processes at time k is described by:

$$\begin{aligned} u(k) = u(k-1) + K_{\mathrm{I}}\left(fal(e(k)) - fal(e(k-1))\right) + K_{\mathrm{II}} \cdot fal(e(k)) \\ + K_{\mathrm{III}}\left(fal(e(k)) - 2fal(e(k-1)) + fal(e(k-2))\right) \end{aligned} \tag{5}$$

Here, K_{II} and K_{III} are respectively represented as integral gain coefficient and differential gain parameter, and their calculation formulas can be expressed as: $K_{\mathrm{II}} = K_{\mathrm{I}} \cdot t_{\mathrm{s}} \cdot T_{\mathrm{i}}^{-1}$ and $K_{\mathrm{III}} = K_{\mathrm{I}} \cdot T_{\mathrm{d}} \cdot t_{\mathrm{s}}^{-1}$.

2.2 PSO Algorithm

The PSO (particle swarm optimization) algorithm is a biologically inspired swarm intelligence method that mimics the collective search behavior of bird flocks foraging. The foraging process of the bird flocks exhibits unpredictable movement behaviors, sometimes suddenly changing their flight trajectory, such as changing direction, dispersing around, gathering with peers, etc. However, scholars have found through research that the entire population of birds maintains consistency throughout the flight process. At the same time, individuals are always in the most suitable position, maintaining the optimal distance for easy collaboration. Ultimately, through the simulation study of this biological group behavior, the basic idea can be summarized as follows: bird flocks search for optimal foraging locations through collaborative behavior and information sharing between the entire group and individuals [27]. In general, the PSO algorithm models the optimization process using a population of particles,

where each particle represents an individual solution within a q -dimensional search space. Furthermore, in order to calculate the optimal position of the entire population, the position and velocity of particles are continuously updated by locating and tracking the particles with the best current position, and compared with parallel particles and spatial particles in the entire population space to find the optimal result (comparing performance indicator functions or fitness functions) [28].

Assuming that in a solution space with a quantity size of Q and a particle dimension of q, there is a particle $\hat{z}_i = [\hat{z}_{i,1}, \hat{z}_{i,2}, \ldots, \hat{z}_{i,q}]$ that represents a vector with q -dimensional position in the i -th particle, here, $i = 1,2,3,\ldots,Q$ In addition, to evaluate the degree of superiority or inferiority of the current particle's position, a function called particle swarm fitness is defined as an optimization performance indicator. Generally, $\hat{z}_i$ represents the current position of a particle in the search space, while $\hat{v}_i = (\hat{v}_{i,1}, \hat{v}_{i,2}, \ldots, \hat{s}_{i,q})$ denotes its velocity or displacement. At this point, in the PSO algorithm, as the current particle is tracked, searched, and updated, a particle $\hat{p}_i = (\hat{p}_{i,1}, \hat{p}_{i,2}, \ldots, \hat{p}_{i,q})$ can be obtained, which is the optimal position located so far. In the end, there exits another particle $\hat{p}_{\mathrm{g}} = (\hat{p}_{\mathrm{g},1}, \hat{p}_{\mathrm{g},2}, \ldots, \hat{p}_{\mathrm{g},q})$, which is the optimal position searched for in the entire population space [27].

As a global optimization strategy, the optimization process of PSO algorithm requires multiple iterations of search. In the particle search space, the flight speed and movement position of the i -th particle in the T -th iteration can be updated according to the following formula [28]:

$$\hat{v}_{i,j}(T+1) = \hat{w} \cdot \hat{v}_{i,j}(T) + c_1 \cdot r_1 \cdot (\hat{p}_{i,j} - \hat{z}_{i,j}) + c_2 \cdot r_2 \cdot (\hat{p}_{g,j} - \hat{z}_{i,j}) \tag{6}$$

$$\hat{z}_{i,j}(T+1) = \hat{z}_{i,j}(T) + \hat{v}_{i,j}(T+1) \tag{7}$$

$$\hat{w} = \hat{w}_{\max} - \frac{\hat{w}_{\max} - \hat{w}_{\min}}{T_{\max}} \times T \tag{8}$$

In the equation, $i = 1,2,\ldots,Q$ and $j = 1,2,\ldots,q$. Here, the variable T indicates the current iteration number for particle searches within the swarm, while $T_{\max}$ defines the maximum allowed iterations. The inertia weight of particle motion is denoted by $\hat{w}$, which varies between minimum inertia weight $\hat{w}_{\min}$ and maximum inertia weight $\hat{w}_{\max}$, The coefficients c_1 and c_2 represent the cognitive and social acceleration factors, and r_1 and r_2 are random values uniformly drawn from 0 to 1. The velocity of the i -th particle after the T -th iteration is given by $\hat{v}_{i,j}(T)$, constrained by $\hat{v}_{\min} \le \hat{v}_{i,j} \le \hat{v}_{\max}$. It is important to note that the particle velocity limits, the minimum velocity $\hat{v}_{\min}$ and the maximum velocity $\hat{v}_{\max}$, define both the possible range for the global optimum and the scale of the fitness function (performance comparison function) within the population.

Optimization of the NLPID control parameters in the batch process is carried out through the PSO algorithm. There are three control parameters that need to be optimized, namely K_{I} , K_{II} and K_{III} [19], [20]. Therefore, the PSO algorithm operates in a 3-dimensional swarm space, with the minimum integral squared error (ISE) serving as the fitness criterion. This is a performance indicator for evaluating the quality of the population space particles (i.e., NLPID control parameters). Its expression is as follows:

$$J(e(k)) = 2^{-1}\sum_{k=1}^{N_k}[y_s(k) - y(k)]^2 = 2^{-1}\sum_{k=1}^{N_k}\left|e(k)\right|^2 \tag{9}$$

In this context, J_{P} is defined as the performance measure corresponding to the tracking error.

2.3 PSO-NLPID Method

This work introduces an NLPID controller optimized using particle swarm optimization, referred to as the PSO-NLPID controller. Therefore, the PSO-NLPID control framework for batch processes can be described as shown in Fig. 2.

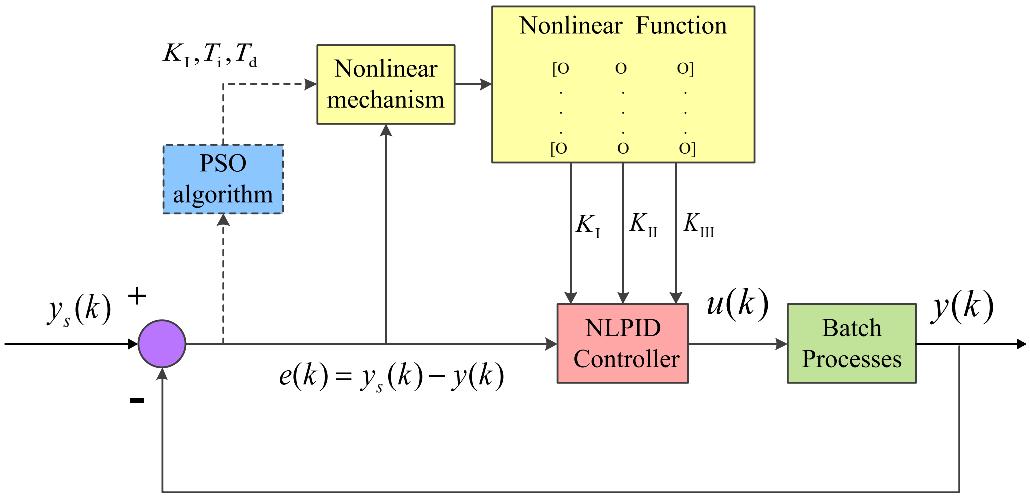

Fig. 2 *Schematic overview of the PSO-NLPID control system for a batch process*

As illustrated in Fig. 2, the PSO algorithm is employed to determine the NLPID controller parameters for batch processes, with the parameter tuning carried out offline and indicated by dashed lines. In the NLPID framework, a nonlinear power function $fal(\cdot)$ is used to adjust the control input according to the tracking error $e(k)$, enabling improved control of the system output. In addition, the control logic of the PSO-NLPID controller should be noted that both a and δ are positive numbers less than 1, and for the nonlinear function $fal(\cdot)$, there is $fal(e(k)) = fal(e(k), a, \delta)$. In this formulation, a is defined as the exponent that characterizes the nonlinear function $fal(\cdot)$, which represents the smoothness of the nonlinear function. As the parameter a increases, the rate of gain will accelerate with the increase of tracking deviation. Furthermore, δ is a threshold parameter for deviation, and its magnitude represents the linear range of the nonlinear function. In this case, selecting an appropriate δ can eliminate the oscillation phenomenon of the controller near the origin; In this research work, it is assumed that the adjustable parameters a and δ are set to 0.75 and 0.1, respectively. Therefore, here are:

$$\begin{aligned}\Delta u(k) &= u(k) - u(k-1) \\ &= K_{\mathrm{I}}\left(fal(e(k),0.75,0.1) - fal(e(k-1),0.75,0.1)\right) + K_{\mathrm{II}} \cdot fal(e(k),0.75,0.1) \\ &+ K_{\mathrm{III}}\left(fal(e(k),0.75,0.1) - 2fal(e(k-1),0.75,1) + fal(e(k-2),0.75,0.1)\right)\end{aligned} \tag{10}$$

In this framework, the parameters a and δ are tuned according to the dynamic behavior of the controlled system. On this basis, the PSO-NLPID controller is employed for batch process control.

Firstly, the NLPID controller parameters (K_{I}, K_{II} and K_{III}) are optimized via PSO (particle swarm optimization). Next, the NLPID controller will utilize the nonlinear fitting mechanism of the exponent function to dynamically adjust the control input to cope with the time-varying dynamic characteristics of batch processes. Finally, the proposed PSO-NLPID method is verified through batch simulation of a representative fermentation process.

The proposed PSO-NLPID approach is implemented through batch simulations of a representative fermentation process, and the overall procedure is illustrated in Fig. 3. The implementation workflow is summarized as follows:

Step 1: A batch fermentation process is selected as the study case, with a total batch duration of 80 hours and a discrete sampling interval of 0.5 hours.

Step 2: An NLPID-based control framework is established for the batch fermentation system.

Step 3: Three key NLPID control gains, namely K_{I}, K_{II}, and K_{III}, are identified as the parameters to be optimized.

Step 4: Particle swarm optimization is employed to tune the selected parameters of **Step 3**, where each particle corresponds to one candidate parameter set and is characterized by its position and velocity in the search space.

Step 5: The performance index J_{p} is defined as the fitness function for evaluating the control performance of the batch fermentation NLPID system.

Step 6: The fitness values of all particles are computed iteratively, and the particle positions $\hat{z}_i$ and velocities $\hat{v}_i$ are updated by comparing individual best values $\hat{p}_i$ and global best values $\hat{p}_g$ until the stopping criterion is satisfied.

Step 7: Upon completion of the optimization process, the optimal performance index $J_{\mathrm{p}} = \hat{p}_g$ is obtained along with the corresponding particle position $\hat{z}_i$, which yields the optimized NLPID parameters.

Step 8: The optimization procedure is terminated.

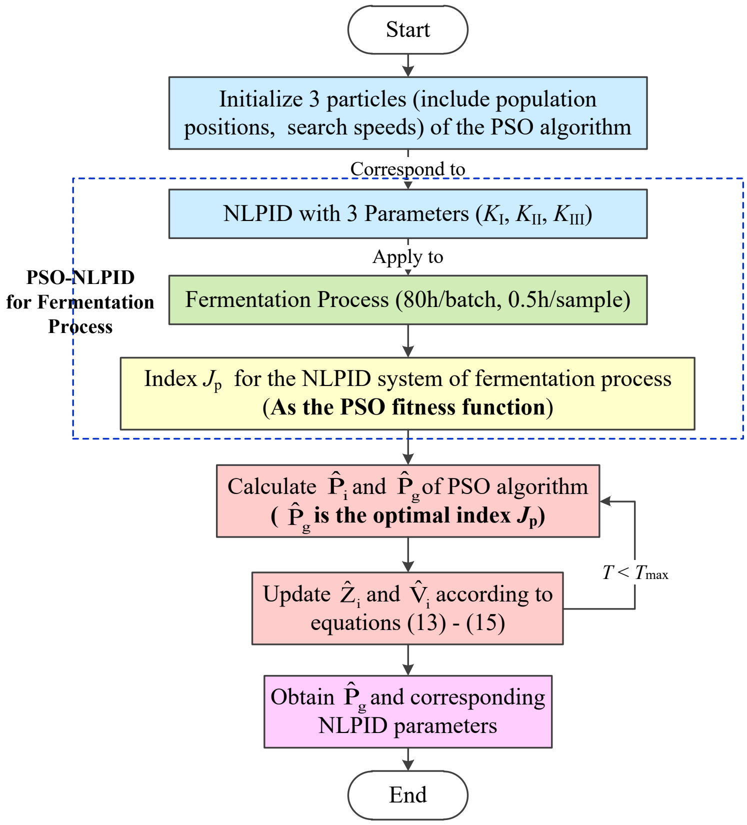

Fig. 3 *Algorithmic flow of the PSO-NLPID control method for batch fermentation processes*

3. Simulation Research and Result Discussion

3.1 Simulation Research

This research mainly employs the dynamic differential equation model of a biological fermentation process to conduct simulation analyses and evaluate the stability and control performance of the proposed PSO-NLPID approach. The simulation platform consisted of a workstation featuring an Intel Core i7-12700H (12th generation) CPU with a base frequency of 2.30 GHz and 16.0 GB of RAM memory. Finally, batch fermentation simulations are completed by generating analysis data via m-file programs developed in MATLAB R2010a. The used biological fermentation process is a typical nonlinear process with time-varying characteristics, and its mechanism mathematical model can be described as the following [29], [30], [31]:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = -DX + \mu X \tag{11}$$

$$\frac{\mathrm{d}S}{\mathrm{d}t} = D(S_{\mathrm{f}} - S) - \frac{\mu X}{Y_{X/S}} \tag{12}$$

$$\frac{\mathrm{d}P}{\mathrm{d}t} = -DP + (\alpha\mu + \beta)X \tag{13}$$

$$\mu = \frac{\mu_{\mathrm{m}}(1 - P/P_{\mathrm{m}})S}{K_{\mathrm{m}} + S + S^2/K_{\mathrm{in}}} \tag{14}$$

where, D and X represent the dilution rate and bacterial concentration of the biological fermentation process, respectively, which are also defined as the control input and system output of the controlled process; Furthermore, S, P, and S_{f} respectively represent the substrate concentration, product concentration, and feed substrate concentration in the biological fermentation process, which are the basic quantities of raw materials for microbial growth. In addition, the bacterial growth rate μ is a very important parameter in the biological fermentation process, which characterizes the inhibitory effect of substrate concentration and product concentration on the biological fermentation process. The biological fermentation process also needs to consider the maximum growth rate μ_{m} of the bacterial cells, the product saturation coefficient P_{m}, the substrate saturation constant K_{m}, the substrate inhibition constant K_{in}, the yield coefficient $Y_{X/S}$ of the bacterial cells to the substrate, and the kinetic parameters of the biological fermentation reaction (α and β). The detailed values and operating conditions of these simulation parameters are derived from the previous research by Henson and Seborg [29].

Usually, dilution rate D is selected as the control variable of the controlled system, that is, the input variable. The bacterial concentration X, substrate concentration S, and product concentration P are all output variables of the controlled process. It should be noted that adjusting the dilution rate D of the fermentation process reasonably can effectively control the bacterial concentration X and ultimately achieve higher quality production [29]. Therefore, selecting bacterial concentration X as the output variable has a reasonable scientific basis [32].

Fig. 4 shows the PSO-NLPID control scheme for the batch process, which controls X by adjusting D. Under the proposed control framework, the batch fermentation operation is divided into four sequential sub-periods, each lasting 20 hours. By applying PSO-NLPID control to the fermentation process, this study investigates the feasibility of hourly control optimization in the presence of parameter variations.

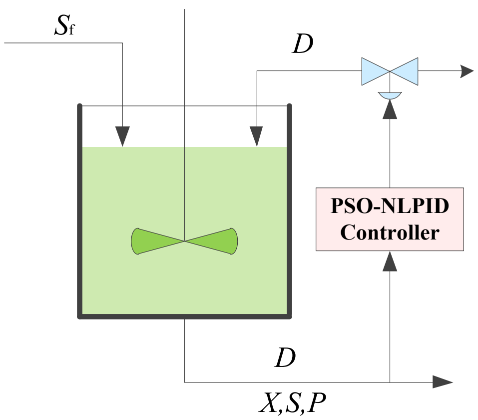

Fig. 4 *The control structural schematic of PSO-NLPID controller in the fermentation process*

3.2 Result Discussion

In order to better solve the control problem of nonlinear intermittent processes, a suitable performance index is used to tune high-quality control parameters, ultimately improving the control effect [32]. The nonlinear control implementation of the PSO-NLPID controller for batch processes involves first selecting a target interval, such as selecting a mass concentration X_k^{set} =4.5~5.5 as the expected target for the batch fermentation process. In this study, the batch cycle lasts 40 hours, while the sampling and discretization times are set to 0.5 hours and 0.05 hours, respectively. The PSO method is then employed to tune the NLPID parameters, and the results are listed in Table 1.

Table 1 *Control parameters of the NLPID controller using PSO algorithm for Xkset=4.5~5.5*

Parameters	Value	Parameters	Value
K_{I}	-0.1955	a	0.75
K_{II}	-0.0221	δ	0.01
K_{III}	0.0422		

This study proposes a PSO-optimized NLPID controller, where the controller parameters are determined via PSO (particle swarm optimization). The tracking behavior for a fermentation process with X_k^{set} =5.0 and fixed parameters is shown in Fig. 5. The feedback input changes of the corresponding NLPID controller are shown in Fig. 6.

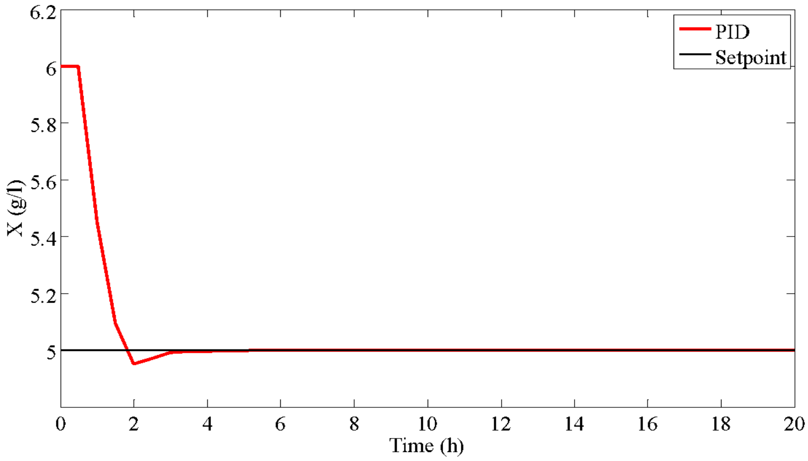

Fig. 5 *Output tracking results of the PSO-NLPID controller under the set-point Xkset=5.0*

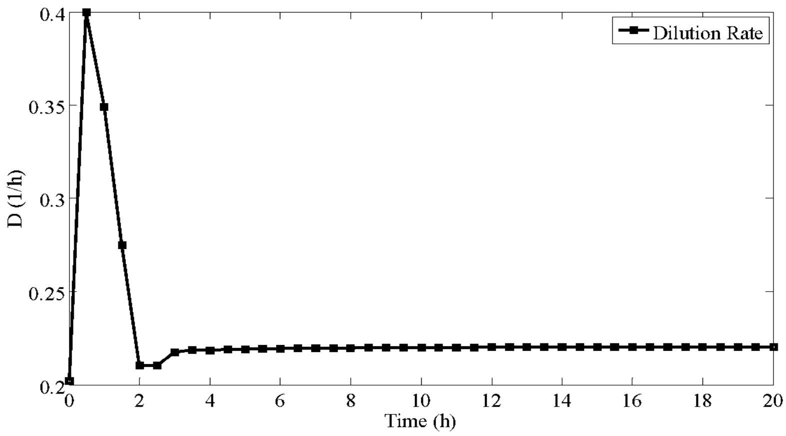

Fig. 6 *Input variations of PSO-NLPID control process in the batch fermentation process Xkset=5.0*

Batch processes often exhibit many problems, such as multi-stage tracking targets, uncertainties and time-varying parameters. For example, during different stages of bacterial growth in biological fermentation processes, the metabolic activity and enzyme activity of microorganisms change over time, which leads to model parameter deviations and uncertainties. The maximum growth rate μ_{m} and substrate yield coefficient $Y_{X/S}$ are two important parameters in the biological fermentation process, and their dynamic changes are unpredictable. Once they occur, they will affect the normal operation of the biological reaction process [32-33]. Previous studies have shown that, if μ_{m} and $Y_{X/S}$ exceed a certain range of values, the controller is unable to adjust them [32]. The time-varying controllable ranges of μ_{m} and $Y_{X/S}$ are shown in Table 2:

Table 2 *The range of μ_{m} and $Y_{X/S}$ with Xkset=4.5~5.5 for the batch fermentation process*

X_k^{set}	4.5	5.0	5.5
μ_{m}	0.48~0.50	0.48~0.50	0.45~0.50
$Y_{X/S}$	0.40~0.60	0.40~0.52	0.40~0.51

Under normal circumstances, the quality of batch process control performance is evaluated using the common ISE index [32], [33]:

$$\mathrm{ISE} = 2^{-1} \cdot \sum_{k=1}^{N_k} \left| e(k) \right|^2 \tag{15}$$

Here, $e(k)$ denotes the value of control error at the k -th moment.

Table 3 *Control performance comparison of PSO-NLPID and other methods for Xkset=4.5~5.5*

Method	Range of Time-varying	ISE Value
PID	μ_{m} =0.48~0.65, $Y_{X/S}$ =0.3~0.55	3.2674
CFMFAC	μ_{m} =0.48~0.65, $Y_{X/S}$ =0.3~0.55	2.1465
PSO-PID	μ_{m} =0.48~0.65, $Y_{X/S}$ =0.3~0.55	1.3411
PSO-NLPID	μ_{m} =0.48~0.65, $Y_{X/S}$ =0.3~0.55	1.1293

Due to inevitable instrument failures, measurement deviations, and process defects in actual industrial processes, these can introduce measurement noise, increase computation time, and lead to a decrease in the control robustness of the controlled process. To verify the effectiveness of the proposed PSO-NLPID control method, traditional PID [29], CFMFAC (i.e. compact form model free adaptive control) [34], PSO-PID [35] were compared with the proposed PSO-NLPID method in the batch fermentation process, considering the uncertainty of parameter variation of μ_{m} at [0.48,0.65] and $Y_{X/S}$ at [0.3,0.55]. Table 3 presents the comparative results of the control performance of the four methods mentioned above. For the parameter time-varying problem in nonlinear batch processes, the results show that the proposed PSO-NLPID method exhibits the lowest ISE value of 1.1293, and its control effect is improved by more than 65% compared to traditional PID, which indicates that PSO-NLPID control has better control performance.

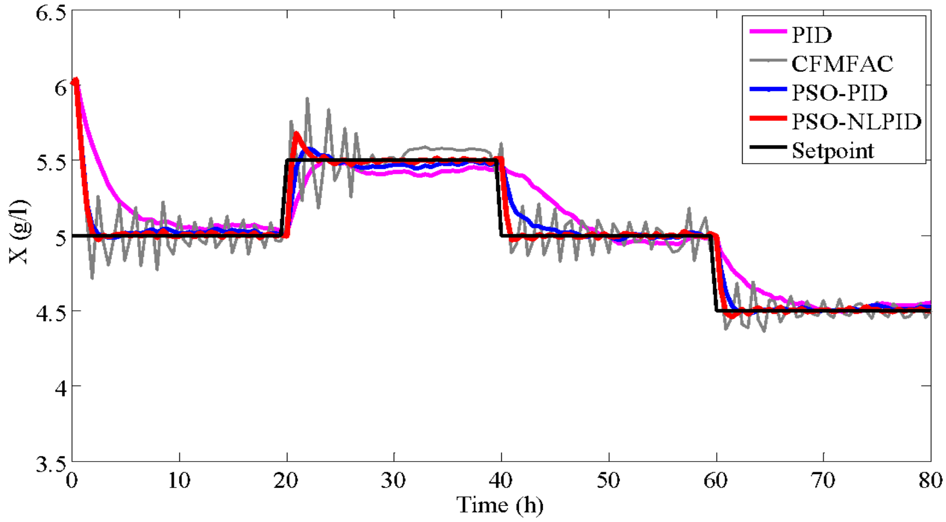

Fig. 7 *Comparative tracking performance of the PSO-NLPID controller and other control strategies in the batch process*

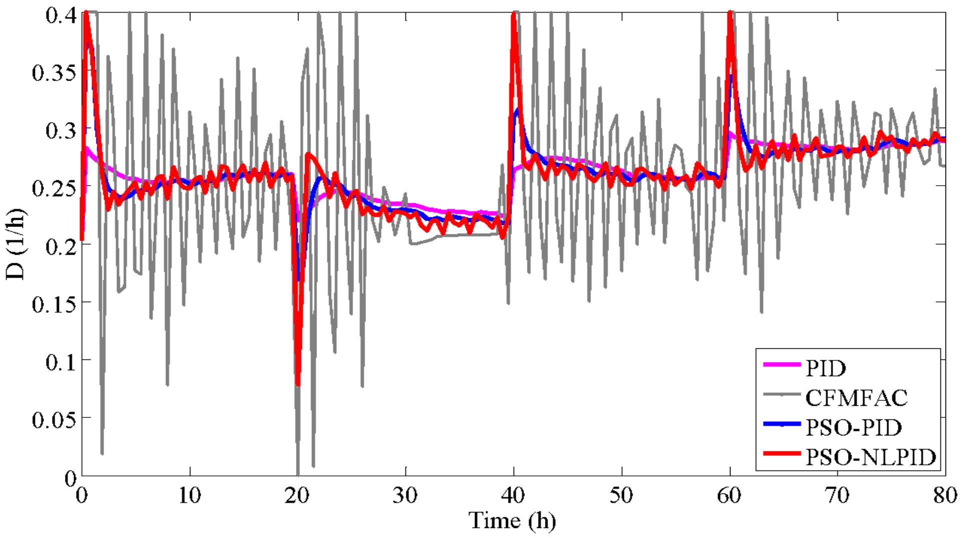

Fig. 8 *The input comparison of PSO-NLPID controller and other methods in the batch process*

The control situation of interference and noise in the controlled process is shown in Figs. 7-10, where Fig. 9 and Fig. 10 respectively describe the changes in process parameters μ_m and $Y_{X/S}$.

In Fig. 7, compared with other methods, the output trajectory controlled by PSO-NLPID is closer to the expected target. The corresponding changes in control input are shown in Fig. 8, indicating that the PSO-NLPID controller can achieve better tracking control by fitting the control input through tracking deviation. Based on the comprehensive situation of Fig. 7 and Fig. 8, it can be analyzed that once the controlled process experiences expected value changes, step disturbances, and noise, the proposed PSO-NLPID approach achieves improved nonlinear control of batch processes by utilizing real-time process information to satisfy the operational demands of individual sub-processes. As the controlled process batch time progresses, the proposed PSO-NLPID control scheme exhibits better control performance. To be summarized, the PSO-NLPID control method demonstrates superior performance in managing nonlinear processes compared to traditional PID, CFMFAC, and PSO-PID controllers. The PSO-NLPID control scheme proposed is a more effective control method for solving the problems of nonlinearity and time-varying parameters in batch processes. This method also provides the possibility to address the computational complexity and scalability of larger systems.

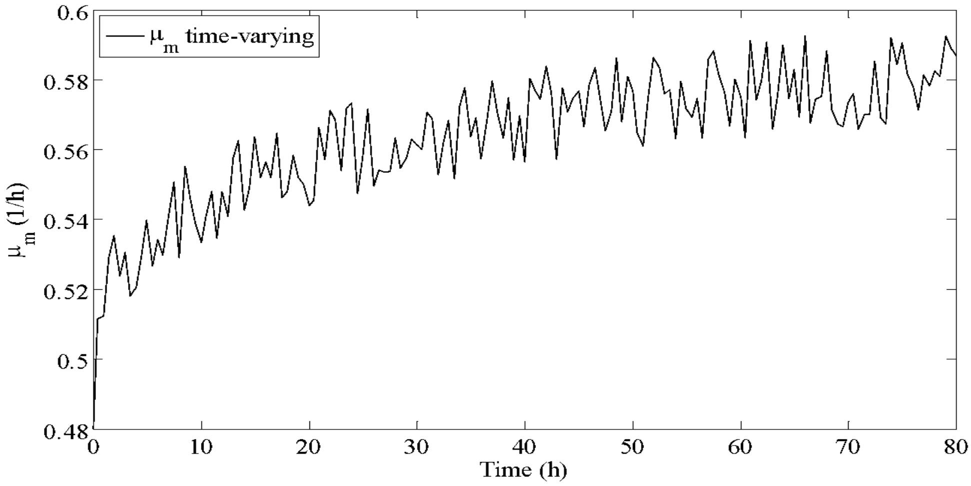

Fig. 9 *Dynamic response of* μ_m *throughout the fermentation process with batch operation*

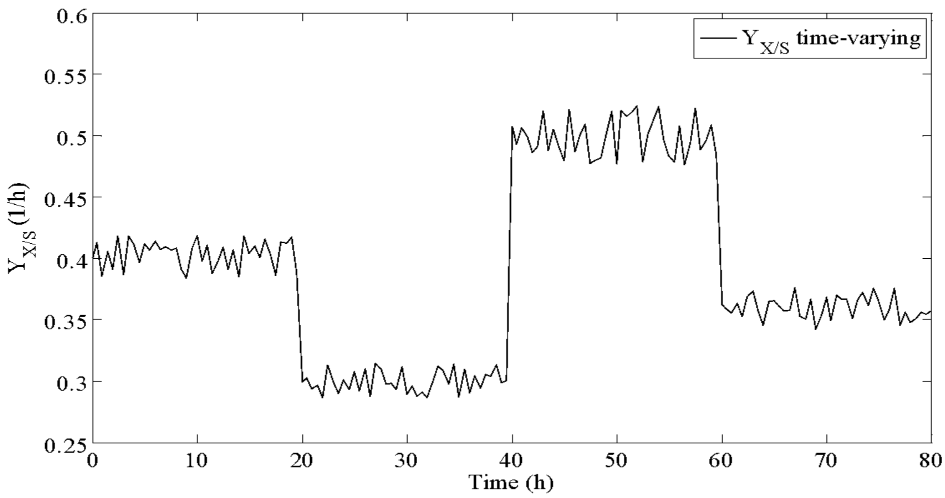

Fig. 10 *Dynamic response of* $Y_{X/S}$ *throughout the fermentation process with batch operation*

4. Conclusion

The challenge of timely establishing accurate mechanism-based models for nonlinear systems highlights the importance of researching practical and efficient nonlinear control techniques for batch operations. The PSO-NLPID control strategy suggested combines a simple configuration with ease of implementation. In response to the nonlinearity and parameter time-varying problems in batch processes, the control input is updated through nonlinear fitting of the deviation signal, resulting in effective tracking performance. In addition, due to the inevitable equipment failures and measurement deviations in industrial processes, it is necessary to consider the control issues of large-scale complex nonlinearity, measurement noise, expected target setting, repeated production operations, and incomplete process information that may exist in batch processes. Consequently, ongoing research is directed toward areas including data-driven feedback controller tuning [36], data-driven control strategies under two-dimensional frameworks [37], adaptive control in the presence of incomplete information [38], iterative learning control combined with active learning mechanisms [39], and dynamic data reconciliation methods [40], [41], [42], etc.

Acknowledgement

This research work was supported by the Fujian Province Young and Middle Aged Research Projects (China) under Grants JAT191447, JAT210878, JAT220715 and Analysis and Research on New Material Technology and Industry in Quanzhou City through Grants 2022GR03.

Conflict of Interest

All authors of this study declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Zhiwen Wang, Shunfeng Ji, Amirul Syafiq Sadun;* ***data collection:*** *Zhiwen Wang, Amirul Syafiq Sadun;* ***analysis and interpretation of results:*** *Zhiwen Wang, Qun Zhang;* ***draft manuscript preparation:*** *Zhiwen Wang, Zheng Chen, Qun Zhuang. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Zhu, J. L., Jia, M. W., Zhang, Y., Zhou, W. H., Deng, H. Y., Liu, Y. (2023). Domain adaptation graph convolution network for quality inferring of batch processes, Chemometrics and Intelligent Laboratory Systems, 243(15):105028-12. https://doi.org/10.1016/j.chemolab.2023.105028

[2] Shi, Y., Hu, X., R., Zhang, Z. M., Xie, L., Xu, W. H., Su, H. Y. (2023). Batch-to-batch optimization for economic performance improvement in batch processes by rational disturbances utilization, Chemical Engineering Research and Design, 190(9): 651-666. https://doi.org/10.1016/j.cherd.2022.12.020

[3] Gao, K. H., Zhou, Y. Q., Lu, J. Y., Gao, F. R. (2024). A selective migration-based improved GPR modeling method for batch process, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 54(5): 3097-3109. https://doi.org/10.1109/TSMC.2024.3353802

[4] Zhou, Y. Q., Gao, K. H., Li, D. W., Xu, Z. H., Gao, F. R. (2021). Data-efficient constrained learning for optimal tracking of batch processes, Industrial & Engineering Chemistry Research, 60(43): 15658–15668. https://doi.org/10.1021/acs.iecr.1c02706

[5] Zhou, Y. Q., Cao, Z. X., Lu, J. Y., Zhao, C. H., Li, D. W., Gao, F. R. (2022). Objectives, challenges, and prospects of batch processes: Arising from injection molding applications, Korean Journal of Chemical Engineering, 39(12): 3179–3189. https://doi.org/10.1007/s11814-022-1294-x

[6] Chandrasekar, A., Baghdassarian, K., Moayedi, F., Abdulhussain, H., Gritsichine, V., Thompson, M. R., Mhaskar P. (2024). Image based Modeling and Control for Batch Processes, Journal of Process Control, 143(2): 103314-11. https://doi.org/10.1016/j.jprocont.2024.103314

[7] Samad, T. (2017). A survey on industry impact and challenges thereof, IEEE Control Systems, 37(1): 17-18. https://doi.org/10.1109/MCS.2016.2621438.

[8] Abramovitch, D. Y. (2019). Thoughts on furthering the control education of practicing engineers, IFAC-PapersOnLine, 52 (9): 85–90. https://doi.org/10.1016/j.ifacol. 2019.08.129

[9] Samad, T., Bauer, M., Bortoff, S., Cairano, S. D., Fagiano, L., et al. (2020). Industry engagement with control research: Perspective and messages, Annual Reviews in Control, 49(9): 1-14. https://doi.org/10.1016/j.arcontrol.2020.03.002

[10] Yoo, H., Byun, H. E., Han, D. H., Lee, J. H. (2021). Reinforcement learning for batch process control: Review and perspectives, Annual Reviews in Control, 52: 108-119. https://doi.org/10.1016/j.arcontrol.2021.10.006

[11] Liu, D., Yang, G. H. (2018). Model-free adaptive control design for nonlinear discrete-time processes with reinforcement learning techniques, International Journal of System Science, 49(11): 2298-2308. https://doi.org/10.1080/00207721.2018.1498557

[12] Borase, R. P., Maghade, D. K., Sondkar, S. Y., Pawar, S. N. (2021). A review of PID control, tuning methods and applications, International Journal of Dynamics and Control, 9(2): 818-827. https://doi.org/10.1007/s40435-020-00665-4

[13] Singh, B., Verma, B., Sharma, S., Padhy, P. K. (2023). Indirect response-based tuning of PID controller with adjustable robustness, Transactions of the Institute of Measurement and Control, 45(6): 3049-3056. https://doi.org/10.1177/01423312231163035

[14] Lee, J. G., Jeong, J., Kim Y. C. (2024). A novel PID controller tuning for desired time response using a pulse response, Journal of Electrical Engineering and Technology, 19(5): 1861-1879. https://doi.org/10.1007/s42835-023-01673-y

[15] Chang, W.D. (2018). A modified PSO algorithm for IIR digital filter modeling, Journal of Circuits, Systems, and Computers, 27(5): 185007-24. https://doi.org/10.1142/S0218126618500731

[16] Imtiaz, U., Jamuar, S., Sahu, J. N., Ganesan, P. (2014). Bioreactor profile control by a nonlinear auto regressive moving average neuro and two degree of freedom PID controllers, Journal Process Control, 24(11): 1761-1777. https://doi.org/10.1016/j.jprocont.2014.09.012

[17] Shi, X. X., Zhang, H., Fan, Z. (2023). Parameter optimization of nonlinear PID controller using RBF neural network for continuous stirred tank reactor, Measurement and Control, 56(9/10): 1835-1843. https://doi.org/10.1177/00202940231189307

[18] Yuliawan, S., Wahyunggoro, O., Setiawan, N. (2021). Kalman filter to improve performance of PID control systems on DC motors, International Journal of Information Technology and Electrical Engineering, 5(3): 96-102. https://doi.org/10.22146/ijitee.64511

[19] Han, J. Q. (1994). Nonlinear PID controller, Acta Automatica Sinica, 20(4): 487-491. https://doi.org/10. 16383 /j . aas . 1994. 04. 015

[20] Han, J. Q. (2009). From PID to active disturbance rejection control, IEEE Transactions on Industrial Electronics, 56(3): 900-907. https://doi.org/10.1109/TIE.2008.2011621

[21] Xiang, K. L., Song, Y. D., Loannou, P. (2025). Nonlinear adaptive PID control for nonlinear systems, IEEE Transactions on Automatic Control, Early Access, 1-8. https://doi.org/10.1109/TAC.2025.3567565

[22] Wang, Z. W., Zeng, F. H., Chen, X. L., Hong, Y. T., Cui, P. (2025). Dynamic data reconciliation to enhance tracking performance of PID control for nonlinear nonaffine systems, Measurement Science and Technology, 36: 046202-046216. https://doi.org/10.1088/1361-6501/adbc0f

[23] Wang, Z. W., Sadun, A. S., Jalaludin, N. A., Jalani, J., Arifin, S. N. H., Mohamed Sunar, N., Fauzi, M. A. (2024). Particle swarm optimization-based model-free adaptive control for time-varying batch processes, International Journal of Automotive and Mechanical Engineering, 21(02): 11264-11276. https://doi.org/10.15282/ijame.21.2.2024.7.0870

[24] Chang, W.D. (2022). An improved particle swarm optimization with multiple strategies for PID control system design, International Journal of Modeling and Optimization, 12(2): 54-60. https://doi.org/10.7763/IJMO.2022.V12.800

[25] Hou, Z. S., Jin, S. T. (2011). A novel data driven control approach for a class of discrete-time nonlinear systems, IEEE Transactions on Control Systems Technology, 19(6): 1549-1558. https://doi.org/10.1109/TCST.2010.2093136

[26] Guo, Y., Hou, Z. S., Liu, S. D., Jin, S. T. (2019). Data-driven model-free adaptive predictive control for a class of MIMO nonlinear discrete-time systems with stability analysis, IEEE Access, 7: 102852-102866. https://doi.org/10.1109/ACCESS.2019.2931198

[27] Eberhart, R. C., Kennedy, J. (1995, October 4-6). A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 39-43. https://doi.org/10.1109/MHS.1995.494215

[28] Kao, C. C., Chuang, C. W., Fung, R. F. (2006). The self-tuning PID control in slider-crank mechanism system by applying particle swarm optimization approach, Mechatronics, 16(8): 513-522. https://doi.org/10.1016/j.mechatronics.2006.03.007

[29] Henson, M. A., Seborg, D. E. (1991). An internal model control strategy for nonlinear system, AIChE Journal, 37(7): 1065-1081. https://doi.org/10.1002/aic.690370711

[30] Radhakrishnan, T. K., Sundarams, S., Chidambaram, M. (1999). Nonlinear control of continuous bioreactors, Bioprocess Engineering, 20(2): 173-178. https://doi.org/10.1007/s004490050577

[31] Venkateswarlu, Ch., Naidu, K. V. S. (2000). Dynamic fuzzy model based predictive controller for a biochemical reactor, Bioprocess Engineering, 23(2): 113-120. https://doi.org/10.1007/s004499900131

[32] Liu. Y., Chen, W. L., Wang, H. Q., Gao, Z. L., Li, P. (2011). Adaptive control of nonlinear time-varying processes using selective recursive kernel learning method, Industrial & Engineering Chemistry Research, 50(5):2773-2780. https://doi.org/10.1021/ie100634k

[33] Gao, Y., Zhao, Z. G., Liu, F. (2018). DMFA-based operation model for fermentation processes, Computers and Chemical Engineering, 109(4): 138-150. https://doi.org/10.1016/j.compchemeng.2017.11.008

[34] Xia, T., Zhang, Z. J., Hong, Z. H., Huang, S. P. (2023). Dynamic data reconciliation to enhance the performance of model free adaptive control, Measurement Science and Technology, 34(6): 065105-12. https://doi.org/10.1088/1361-6501/acbc92

[35] Wang, Z. W., Chen, C. C., Chen, X. L., Li, D. G., Zeng, F. H. (2021, May 14-16). Adaptive PID control for time-varying fermentation processes, IEEE 10th Data Driven Control and Learning Systems Conference, 643-648. https://doi.org/10.1109/DDCLS52934.2021.9455651

[36] Lin, N., Chi, R. H., Liu, Y., Hou, Z. S., Huang, B. (2023). Data-driven set-point tuning of model-free adaptive control, International Journal of Robust and Nonlinear Control, 33(13): 7667-7686. https://doi.org/10.1002/rnc.6788

[37] Chi, R. H., Li, H. Y., Shen, D., Hou, Z.S., Huang, B. (2023). Enhanced P-type control: Indirect adaptive learning from set-point updates, IEEE Transactions on Automatic Control, 2023,60(3): 1600-1613. https://doi.org/10.1109/TAC.2022.3154347

[38] Shen, D. (2018). Iterative learning control with incomplete information: a survey, IEEE/CAA J. of Autom. Sinica, 5(5): 885-901. https://doi.org/10.1109/JAS.2018.7511123

[39] Chen, T., Liu, Y., Chen, J. H. (2013). An integrated approach to active model adaptation and on-line dynamic optimisation of batch processes, Journal of Process Control, 23(10): 1350-1359. https://doi.org/10.1016/j.jprocont.2013.09.010

[40] Zhu, W.W., Zhang, Z. J., Liu, Y. (2022). Dynamic data reconciliation for improving the prediction performance of the data-driven model on distributed product outputs, Industrial & Engineering Chemistry Research, 61(51): 18780-18794. https://doi.org/10.1021/acs.iecr.2c02536

[41] Zhu, W. W., Jia, M. W., Zhang, Z. J., Liu, Y. (2024). Dynamic data reconciliation for enhancing the performance of kernel learning soft sensor models considering measurement noise, Chemometrics and Intelligent Laboratory Systems, 246(1-2): 105083 https://doi.org/10.1016/j.chemolab.2024.105083

[42] Zhu, W. W., Zhu, J. L., Yang, Q. M., Liu, Y., Zhang, Z. J. (2024). Dynamic data reconciliation for enhancing the prediction performance of long short-term memory network, Measurement Science and Technology, 35(11): 116147. https://doi.org/10.1088/1361-6501/ad70d2