INTERNATIONAL JOURNAL OF AUTOMOTIVE AND MECHANICAL ENGINEERING
ISSN: 2229-8649 e-ISSN: 2180-1606
VOL. 21, ISSUE 2, 2024, 11263 – 11275
DOI: https://doi.org/10.15282/ijame.21.2.2024.7.0870

RESEARCH ARTICLE

Particle Swarm Optimization-Based Model-Free Adaptive Control for Time-Varying Batch Processes

Z. Wang[1], A. S. Sadun[1*], N. A. Jalaludin[1], J. Jalani[2], S. N. H. Arifin[3], N. Mohamed Sunar[3,4], M. A. Fauzi[5]

[1]Department of Electrical Engineering Technology, Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia, Pagoh Higher Education Hub, 84600 Pagoh, Muar, Johor, Malaysia
[2]Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400 Batu Pahat, Johor, Malaysia
[3]Research Center for Soft Soil, Institute of Integrated Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400 Batu Pahat, Johor, Malaysia
[4]Department of Civil Engineering Technology, Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia, Pagoh Higher Education Hub, 84600 Pagoh, Muar, Johor, Malaysia
[5]Faculty of Industrial Management, Universiti Malaysia Pahang Al-Sultan Abdullah, Persiaran Tun Khalil Yaakob, 26300, Kuantan, Pahang, Malaysia

ABSTRACT - The batch process is a production process with strong nonlinearity, which usually suffers from time-varying parameters and uncertainty of disturbances. Concerning the mentioned problems, this study proposes to investigate the application of the particle swarm optimization-based model-free adaptive control (PSO-MFAC) method for time-varying batch processes. Model-Free Adaptive Control (MFAC) is a data-driven control method, which is one of the promising methods to solve the nonlinear process. Firstly, a Full Form Dynamic Linearization Model-Free Adaptive Control method has been adopted for the control of batch processes. Further, considering that the adopted model-free adaptive control involves seven control parameters, such as cognitive scaling factor (φ_1), social scaling factor (φ_2), inertia weight (φ_3), learning rate (η), control parameter update rate (μ_C), exploration rate (ρ) and learning rate (λ) for MFAC obtained by a particle swarm optimization (PSO) algorithm in combination with a criterion function performance index. Finally, by comparing it with the existing methods, a typical batch fermentation was applied to verify that PSO-MFAC had a good control effect. The findings indicate that the PSO-MFAC controller exhibits a preference for exploiting the optimal option due to its φ_3 value less than 0.1. The efficacy and feasibility of the PSO-MFAC control effect have been proven by obtaining the lowest integral square error (ISE) value of 1.1192 regarding the nonlinearity of the batch process due to time-varying challenges.

ARTICLE HISTORY
Received : 13th Dec. 2023
Revised : 03rd Apr. 2024
Accepted : 29th Apr. 2024
Published : 20th June 2024

KEYWORDS
Adaptive control
Batch process
Human-robot interaction
Model-free
Nonlinear process
Particle swarm optimization

1.0 INTRODUCTION

Batch processes are frequently employed in industries characterized by high-quality standards, diversified product offerings, and relatively low production volumes, such as fine chemicals, biopharmaceuticals, and food fermentation, in contrast to continuous processes commonly found in large-scale manufacturing facilities [1-3]. In a continuous process, the operation typically remains at the optimal economic point; in contrast, the operational conditions of a batch process change dynamically from beginning to end. As a result of this dynamic change, batch operations are more flexible than continuous operations, allowing operating conditions and cycle duration to be modified to correspond with actual production demand [4]. However, the strong nonlinearity inherent in the batch process is characterized by both time-varying dynamics and non-stationary operating conditions, involving transition processes with variable objectives and covering transient phases of the operating range [5]. These issues bring great challenges to the control of batch processes.

Although the control theory of industrial processes has been vigorously developed in recent decades, and ever-changing control techniques are emerging, most of the control methods are only applicable to continuous processes, while they are powerless for batch processes, which are more complex in the system [6-7]. Research indicates that in batch process control, different strategies are used to prevent interference caused by changing model parameters and physicochemical processes. These strategies include adapting model parameters, correction conditions, and input variables [8-11]. For the present, most of the studies are still limited to simple proportional integral derivative (PID) control [12], neural network control [13] (NN), and model predictive control [14] (MPC). Among these methods, traditional PID controllers have difficulty overcoming the complexity and nonlinearity of industrial processes [15]. Furthermore, control methods such as NN and MPC make the actual process modeling more difficult, and model training requires a long period to ensure the reliability of the trained model [16].

For this study, data types vary based on research objectives and methods. Relevant data include experimental (e.g., sensor readings), process (e.g., task details), control (e.g., strategy info), simulation (e.g., model-generated data), optimization (e.g., PSO parameters), human-related (e.g., behavior), system identification, performance metrics, and historical data. The data selection and pre-processing of data are done carefully and in detail, ensuring ethical compliance.

***CORRESPONDING AUTHOR | A. S. Sadun** | ✉ amirul@uthm.edu.my

A data-driven control method is effective for industrial processes with nonlinear and time-varying problems [17]. While PID control is widely used in production due to its simple structure and ease of implementation, it faces challenges with nonlinear, time-varying, and uncertain systems. In such cases, regulating its parameters can be difficult, leading to less satisfactory outcomes [18]. Model-free adaptive control (MFAC) is a type of data-driven control method that does not rely on a mechanistic or process model of the system. Instead, it uses the real-time input/output (I/O) data from the controlled system to design and implement adaptive control. This approach has been successful in achieving effective control performances. [19]. Recently, MFAC has gained significant popularity in the syngas manufacturing industry [20], intelligent transportation [21], motor systems [22], and so on.

Compact form dynamic linearization model-free adaptive control (CFDL-MFAC), partial form dynamic linearization model-free adaptive control (PFDL-MFAC), and full form dynamic linearization model-free adaptive control (FFDL-MFAC) are the three different forms of the MFAC control method [23]. Dynamic linearization excels due to its simplicity, adaptability to time-varying dynamics, model-free nature, ease of implementation, integration with optimization, and analytical insights. These traits align with the study's focus on model-free adaptive control in a dynamic environment, making it more suitable than alternative approaches for the specified research. First, for CFDL-MFAC, the structure is the simplest, considering only one feedback control for tracking deviation, which is equivalent to the time-varying deviation control term for tracking deviation; in addition, PFDL-MFAC enhances the previous approach by taking into account the dynamic and time-varying correlation between the change in the system's output in the next moment and the change in the inputs within a fixed-length sliding time window in the present moment; and finally, FFDL-MFAC, on the basis of the second MFAC form, also considers the effect of the comprehensive changes in the control inputs and the whole changes in the system outputs using a fixed-length sliding time window, then analyze the changes of system outputs in the next moment based on the present moment [23]. Given that the processing system is highly nonlinear, the first two forms do not always yield the desired satisfactory results [18].

Particle swarm optimization (PSO) is a global optimization algorithm that is effective in solving complex optimization problems involving nonlinear, non-differentiable, and multi-peaked functions. It has gained significant attention and application due to its simplicity, ease of implementation, minimal adjustable parameters, and favorable consequences [24]. Recently, researchers have utilized PSO to optimize the parameters of PID controllers, aiming to enhance the control efficacy [12]. However, such improved MFAC controllers are still rarely used for batch processes. Therefore, in order to better overcome the strong nonlinearity that batch processes have, the study firstly implements the third form of MFAC mentioned above, i.e., FFDL-MFAC, which will still be referred to as MFAC in the subsequent papers for the sake of convenience, secondly, considering a criterion function performance index that incorporates the squared accumulation of the tracking error and the input difference, the PSO method is employed to optimize the MFAC controller using seven initial values for the regulatory parameters. Ultimately, the proposed PSO-MFAC is utilized to control the intra-batch time-varying process.

Therefore, the objective of this study is to examine the efficacy of the PSO-based MFAC (PSO-MFAC) method in addressing the nonlinearity of the batch process with time-varying particularities. The purpose of developing data-driven control approaches for time-varying batch processes holds substantial practical importance due to the inherent challenges in effectively modeling and controlling such processes.

2.0 METHOD AND MATERIALS

2.1 PSO-MFAC Control for Batch Processes

To illustrate the general state form of batch processes, the study utilized a single-input-single-output (SISO) discrete nonlinear system, represented by Equation (1) [25-26]:

$$\begin{aligned} y(k+1) = f(&y(k), y(k-1), \ldots, y(k-\mathrm{ny}), \\ &u(k), u(k-1), \ldots, u(k-\mathrm{nu}), \\ &v(k), v(k-1), \ldots, v(k-\mathrm{nv})) \end{aligned} \tag{1}$$

where the state function denotes a nonlinear function describing the batch process, a dynamic system that varies with moment k; $y(k)$, $u(k)$, $v(k)$ are the output variable, the input variable (also known as control variables), and the process disturbances (mainly including the model parameter disturbances, the process noise and measurement noise are considered distinct components, with the disturbances presumed to represent extra factors. *ny*, *nu*, and *nv* are integers denoted as *ny, nu, nv* ∈ {1,2, 3…}.

As the object of study, batch processes usually show the characteristics of the desired goal staged. According to the production nature of the batch process, set its batch time as t_f, sample every t_s time, the entire batch cycle of the batch process can be separated into $\widetilde{N}$ sub-process intervals, i.e. $\widetilde{N} = t_f / t_s$, $k = 1,2, \ldots, \widetilde{N}$, the entire cycle of the batch process involves using the MFAC controller to modify the input variables in a rational manner, hence controlling the output variables to achieve the optimum production capacity and satisfy the needs of the end product. The PSO-MFAC parameters were carefully chosen to reflect the complexity and dynamic nature of batch processes, particularly when considering a human-robot interaction environment.

Initialize the batch process with the provided input and output circumstances $[u(0),y(0)]$ for computational convenience. The structure block diagram of a SISO batch process MFAC control system is depicted in Figure 1.

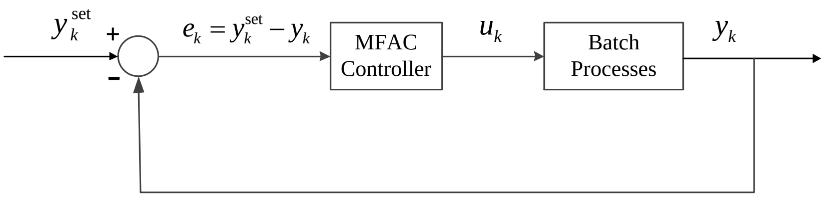

Figure 1. MFAC control Framework for batch processes

At the beginning of the study, the following assumptions were made for the system of batch processes, i.e., Equation (1):

Assumption 1: The continuity of the partial derivatives of the nonlinear state function *f(·)* with respect to the system inputs *u(k)* is confirmed.

Assumption 2: The batch process system Eq. (1) fulfills the criteria of the condition of generalized Lipschitz, i.e., for any moment k and $\Delta u(k) \neq 0$, there are:

$$\Delta y(k+1) = \bar{\varphi} \cdot |\Delta u(k)| \tag{2}$$

where, $\bar{\varphi} > 0$ is a constant.

According to the literature [23], under the conditions of Assumption 1 and Assumption 2, the controlled system described in Equation (1) can be transformed into a Full Form Dynamic Linearization (FFDL) data model that requires only I/O data, namely.

$$y(k+1) = y(k) + \varphi_{f,Ly,Lu}^T(k) \cdot \Delta H_{Ly,Lu}(k) \tag{3}$$

where the subscript f of the pseudo partial derivative (PPD) $\varphi_{f,Ly,Lu}^T(k)$ denotes full, Ly and Lu are pseudo-orders, $0 \leq Ly \leq ny,\ 1 \leq Lu \leq nu$. The transformation of the FFDL data model into a PFDL data model occurs when Lu is equivalent to 0 and Lu is equivalent to L. Similarly, the FFDL data model transforms into a CFDL data model when Lu is equivalent to 0 and Lu is equivalent to 1.

Define the process parameters: pseudo-partial derivatives $\varphi_{f,Ly,Lu}^T(k)$ and combinatorial vectors $\Delta H_{Ly,Lu}(k)$, which are combinatorial vectors of output variations $\Delta y(k)$ and input variations $\Delta u(k)$ with the following sequence of vectors:

$$\varphi_{f,Ly,Lu}^T(k) = [\varphi_1(k), \varphi_2(k), \ldots, \varphi_{Ly}(k), \varphi_{Ly+1}(k), \ldots, \varphi_{Ly+Lu}(k)]^T \tag{4}$$

$$\Delta H_{Ly,Lu}(k) = [\Delta y(k), \Delta y(k-1), \ldots, \Delta y(k-Ly+1), \Delta u(k), \Delta u(k-1), \ldots, \Delta u(k-Lu+1)]^T \tag{5}$$

Next, consider the following FFDL-MFAC control input criterion function:

$$J_1(\Delta u(k)) = [yr(k+1) - y(k+1)]^2 + \lambda \cdot [\Delta u(k)]^2 \tag{6}$$

where J_1 is the control input objective function, $yr(k+1)$ is the process output expected value, $e(k+1)$ is the process tracking error value, and $\lambda > 0$ is a weighting constant. In addition, there are the following definitions:

$$e(k+1) = yr(k+1) - y(k+1) \tag{7}$$

$$\Delta u(k) = u(k) - u(k-1) \tag{8}$$

Bringing Equation (3) into Equation (6), first solving for the derivative of $\Delta u(k)$ and then making it zero, the learning law of the batch process FFDL-MFAC controller, i.e., the feedback expression for the input variable $u(k)$ is shown below:

$$\begin{aligned}\Delta u(k) = &\frac{\varphi_{Ly+1}(k) \cdot \rho_{Ly+1} \cdot [yr(k+1) - y(k+1)]}{\lambda + [\varphi_{Ly+1}(k)]^2} \\ &- \frac{\varphi_{Ly+1}(k) \cdot \sum_{l=1}^{Ly} \cdot \rho_l \cdot \varphi_l(k) \cdot \Delta y(k-l+1)}{\lambda + [\varphi_{Ly+1}(k)]^2} \\ &- \frac{\varphi_{Ly+1}(k) \cdot \sum_{l=Ly+2}^{Ly+Lu} \rho_l \cdot \varphi_l(k) \cdot \Delta u(k+Ly-l+1)}{\lambda + [\varphi_{Ly+1}(k)]^2} \\ &= \frac{\varphi_{Ly+1}(k) \cdot \rho_{Ly+1}}{\lambda + [\varphi_{Ly+1}(k)]^2} e(k)\end{aligned} \tag{9}$$

$$- \frac{\varphi_{Ly+1}(k) \cdot \sum_{l=1}^{Ly} \cdot \rho_l \cdot \varphi_l(k) \cdot \Delta y(k-l+1)}{\lambda + [\varphi_{Ly+1}(k)]^2}$$ (9) cont.

$$-\frac{\varphi_{Ly+1}(k) \cdot \sum_{l=Ly+2}^{Ly+Lu} \rho_l \cdot \varphi_l(k) \cdot \Delta u(k+Ly-l+1)}{\lambda + [\varphi_{Ly+1}(k)]^2}$$

The step factor $\rho_l > 0$, where l is an integer ranging from 1 to L, is incorporated to enhance the versatility of the control method. The pseudo-order of the FFDL-MFAC controller is specifically set as Ly=1 and Lu=2, and thus $\varphi_{f,Ly,Lu}^T(k) = [\varphi_1 \ \varphi_2 \ \varphi_3 \ ]^T$ is obtained.

Further, consider the following FFDL-MFAC partial derivative PPD estimation criterion function:

$$\begin{gathered} J_2(\hat{\varphi}_{f,Ly,Lu}^T(k)) = [y(k) - y(k-1) \\ -\hat{\varphi}_{f,Ly,Lu}^T(k) \cdot \Delta H_{Ly,Lu}(k-1)]^2 \\ +\mu_C \cdot [\hat{\varphi}_{f,Ly,Lu}^T(k) - \hat{\varphi}_{f,Ly,Lu}^T(k-1)]^2 \end{gathered} \tag{10}$$

where J_2 is the partial derivative estimation objective function and $\mu_C > 0$ is a weight constant. In turn, the online estimation algorithm of $\hat{\varphi}_{f,Ly,Lu}^T(k)$ can be known:

$$\begin{gathered} \hat{\varphi}_{f,Ly,Lu}^T(k) = \hat{\varphi}_{f,Ly,Lu}^T(k-1) + \frac{\eta \cdot \Delta H_{Ly,Lu}(k-1)}{\mu_C + [\Delta H_{Ly,Lu}(k-1)]^2} \\ \cdot [y(k) - y(k-1) - \hat{\varphi}_{f,Ly,Lu}^T(k-1) \cdot \Delta H_{Ly,Lu}(k-1)] \\ = \hat{\varphi}_{f,Ly,Lu}^T(k-1) + \frac{\eta \cdot \Delta H_{Ly,Lu}(k-1)}{\mu_C + [\Delta H_{Ly,Lu}(k-1)]^2} \\ \cdot [\Delta y(k) - \hat{\varphi}_{f,Ly,Lu}^T(k-1) \cdot \Delta H_{Ly,Lu}(k-1)] \end{gathered} \tag{11}$$

In order to make the control algorithm more general, a step factor $\eta \in (0,2]$ is introduced. In addition, there exists a PPD parameter reset algorithm:

$$\begin{gathered} \text{If } \left|\hat{\varphi}_{f,Ly,Lu}^T(k)\right| \le \varepsilon \\ \text{Or } \left|\Delta H_{Ly,Lu}(k-1)\right| \le \varepsilon \\ \text{Or } sign(\hat{\varphi}_{f,Ly,Lu}^T(k)) \ne sign(\hat{\varphi}_{f,Ly,Lu}^T(1)) \\ \text{Then } \hat{\varphi}_{f,Ly,Lu}^T(k) = \hat{\varphi}_{f,Ly,Lu}^T(1) \end{gathered} \tag{12}$$

$\hat{\varphi}_{f,Ly,Lu}^T(1)$ is the initial value of the PPD, and the algorithm reset mechanism is introduced to make the PPD estimation algorithm more capable of tracking time-varying parameters.

2.2 PSO Algorithm

PSO algorithm is a class of biological group-inspired intelligent algorithms that originated from the study of bird flock foraging behavior. Researchers have found that birds suddenly change information to trajectories during the foraging process, such as changing direction, spreading out, gathering quickly, etc., and their behavior is unpredictable, but the whole group is always consistent; at the same time, the individuals have always maintained the optimal distance from each other; through the behavioral study of this kind of biological groups, it can be observed that the primary idea is to discover the ideal solution through the interactive behaviors of the individuals in the flock and the sharing of information to find the optimal solution [27]. Within the PSO algorithm, the population is defined as a collection of whole entities, while particles refer to individual entities. To achieve the best possible outcome for the overall population, the current ideal particle is identified and monitored among the particles in the search space of dimension d. The orientation and momentum of the particle are continuously revised, and the optimal solution is sought with the parallel particles and the spatial particles throughout the entire plenary space (comparing the optimization performance index function or known as the fitness function) [28].

Assuming that there are N d-dimensional space particles, a solution space of size N can be set. $z_i = (z_{i,1}, z_{i,2}, \dots, z_{i,d})$ can be used to represent the first particle of the d-dimensional position vector, where i =1, 2..., N; at the same time, a performance index is expressed to determine the intensity of optimization of the particle's positional superiority or inferiority, which is called the particle swarm fitness evaluation function. Then, it can be known that the current position of the optimized spatial particle is z_i ; The particle's current flight speed or distance is denoted as $s_i = (s_{i,1}, s_{i,2}, \dots, s_{i,d})$. $p_i = (p_{i,1}, p_{i,2}, \dots, p_{i,d})$ reflects the best position found by the particle to this point. Ultimately, the overall best position found by the entire particle swarm in the current moment can be written as $p_g = (p_{g,1}, p_{g,2}, \dots, p_{g,d})$ [27].

The formula is used to update the position (z_i) and the velocity (s_i) of the space particle, in the i (dimensional space), in the T (iteration of the spatial search). [28]:

$$s_{i,j}(T+1) = \tilde{\lambda} s_{i,j}(T) + c_1 r_1 (p_{i,j} - z_{i,j}) + c_2 r_2 (p_{g,j} - z_{i,j}) \tag{13}$$

$$z_{i,j}(T+1) = z_{i,j}(T) + s_{i,j}(T+1) \tag{14}$$

$$\tilde{\lambda} = \tilde{\lambda}_{max} - \frac{\tilde{\lambda}_{max} - \tilde{\lambda}_{min}}{T_{max}} \times T \tag{15}$$

where $i = 1,2,\ldots,N$, $j = 1,2,\ldots,d$; T represents the number of iterations in the group particle search space, while T_{max} represents the maximum number of iterations. $\tilde{\lambda}$ is the coefficient for the inertia weight of space particle flight, with $\tilde{\lambda}_{min}$ being the minimum value and $\tilde{\lambda}_{max}$ being the maximum value. c_1 and c_2 are the acceleration factors for space particle flight. r_1 and r_2 are random numbers between 0 and 1. $s_{i,j}(T)$ represents the acceleration of the first particle after the second iteration, with $s_{min} < s_{i,j} < s_{max}$. In addition, s_{max} and s_{min} are the maximum and minimum update rates of the spatial population particles, specify the size of the particle population's global ideal position, and specify the population fitness assessment function or optimization performance indicator function.

The PSO algorithm is used to optimize the batch process's MFAC settings. The adopted FFDL-MFAC has seven parameters, which are cognitive scaling factor (φ_1), social scaling factor (φ_2), inertia weight (φ_3), learning rate (η), control parameter update rate (μ_C), exploration rate (ρ) and learning rate for MFAC (λ) [23]. It should be noted that the "cognitive scaling factor (φ_1), social scaling factor (φ_2), and inertial scaling factor (φ_3)" in MFAC are one of the control parameters, which mainly act on the controlled object (such as fermentation process) to achieve tracking control; The inertia weight coefficients ($\tilde{\lambda}$) and acceleration factors (c1 and c2) of the PSO method are only parameters of the search algorithm itself and only apply to the optimization-solving process. In addition, based on the influence of parameters in the formula, the range of values for the above seven parameters is marked under the corresponding formula. By randomly selecting any number near the value range, the powerful optimization ability of the PSO algorithm can easily obtain the optimal parameter values. Therefore, the particle population of the PSO algorithm is set as a seven-dimensional space, and the addition of the tracking error squared cumulative and input difference is chosen as the minimum fitness evaluation function, which is the performance index to discriminate the superiority of the population particles (control parameters), as follows.

$$\begin{aligned} J_3(e(k)) &= 2^{-1}\sum_{k=1}^{N_k}[y_r(k) - y(k)]^2 + |u(k) - u(k-1)| \\ &= 2^{-1}\sum_{k=1}^{N_k} e^2(k) + |\Delta u(k)| \end{aligned} \tag{16}$$

where J_3 is the target tracking error performance metric.

2.3 PSO-MFAC Controller

In this section, a particle swarm optimization-based FFDL-MFAC controller, PSO-MFAC controller, is introduced for the batch process, the adaptive process of which is shown in Figure 2.

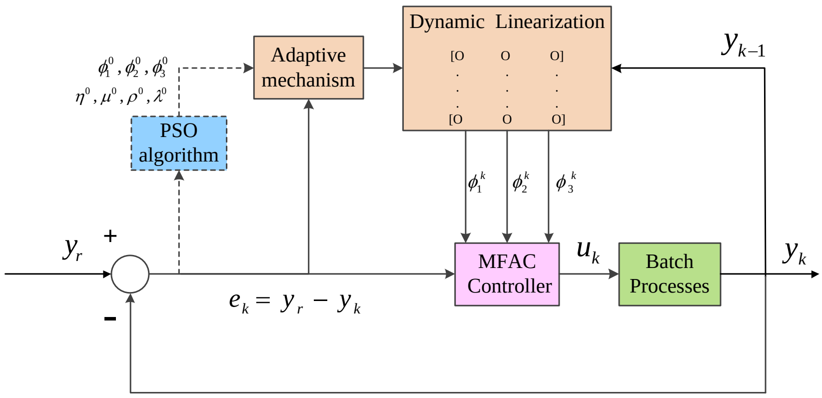

Figure 2. PSO-MFAC controller for the batch process

Figure 2 illustrates the PSO algorithm is used for the off-line optimization of the MFAC controller parameters for the batch process, indicated by the dashed lines; the adaptive mechanism is implemented by constantly updating the pseudo partial derivatives φ_1, φ_2, φ_3 by means of a full-format dynamic linearization method. In addition to this, for the PSO-MFAC controller sense, the controller selects a set of pseudo-orders Ly = 1 and Lu = 2; furthermore, as the control parameters of the MFAC, on the one hand, the parameter values of the optimized φ_1, φ_2, φ_3 are different; on the other hand, as the pseudo-partial derivatives, $\varphi_{f,Ly,Lu}^T(k)$ =[φ_1 φ_2 φ_3]^T will be updated and learned accordingly with the time-varying dynamics of the system of the batch process, and thus, it can be made $\rho_1 = \rho_2 = \rho_3 = \rho \in (0,1]$. Thus, there are:

$$\Delta u(k) = \frac{\varphi_2(k)\cdot\rho_2}{\lambda + [\varphi_2(k)]^2}e(k) - \frac{\varphi_2(k)\cdot\rho_1\cdot\varphi_1(k)\cdot\Delta y(k)}{\lambda + [\varphi_2(k)]^2} - \frac{\varphi_2(k)\cdot\rho_3\cdot\varphi_3(k)\cdot\Delta u(k-1)}{\lambda + [\varphi_2(k)]^2} \tag{17}$$

$$= \frac{\varphi_2(k)\cdot\rho}{\lambda+[\varphi_2(k)]^2} e(k) - \frac{\varphi_2(k)\cdot\rho\cdot\varphi_1(k)\cdot\Delta y(k)}{\lambda+[\varphi_2(k)]^2} - \frac{\varphi_2(k)\cdot\rho\cdot\varphi_3(k)\cdot\Delta u(k-1)}{\lambda+[\varphi_2(k)]^2}$$

And the updating law of $\varphi^T_{f,Ly,Lu}(k) = [\varphi_1\ \varphi_2\ \varphi_3]^T$ can be directly referred to Eq. (11) in this paper. Controlling batch operations with the suggested PSO-MFAC controller; The PSO algorithm first initializes the control parameters of the MFAC controller, such as φ_1、φ_2、φ_3、η、μ_C、$\rho$and λ. Second, the MFAC's dynamic linearization adaptive mechanism continually updates the φ_1, φ_2 and φ_3 at different points in the batch process to adjust to the time-varying dynamics of the system. Finally, batch simulation of a typical fermentation process is used to investigate the control effect of the proposed PSO-MFAC.

The implementation flowchart of the PSO-MFAC method used for batch simulation of the fermentation process is shown in Figure 3. The main steps of its implementation are:

Step 1: Take the fermentation tank as the object, with a batch running for 80 hours and a cycle of 0.5 hours.

Step 2: Build an MFAC control system for the fermentation process.

Step 3: Select the 7 MFAC control parameters that need to be optimized (ϕ_1, ϕ_2, ϕ_3, η, μ, ρ, λ);

Step 4: Use the PSO algorithm to optimize the seven parameters in Step 3. PSO initializes the seven particle parameters corresponding to them. The particle information includes the position of the particle itself, the positions of the particle group, and the search speeds of these particles.

Step 5: Select J_3 as the PSO algorithm optimization fitness function for the fermentation process MFAC system.

Step 6: By calculating the fitness function value of PSO optimization, continuously compare p_i and p_g to update the particle's position z_i and velocitys_i until the end of the iteration.

Step 7: Obtain the optimal indicator value $J_3 = p_g$ and the particle positions (7 parameter values of MFAC) corresponding to p_g;

Step 8: End.

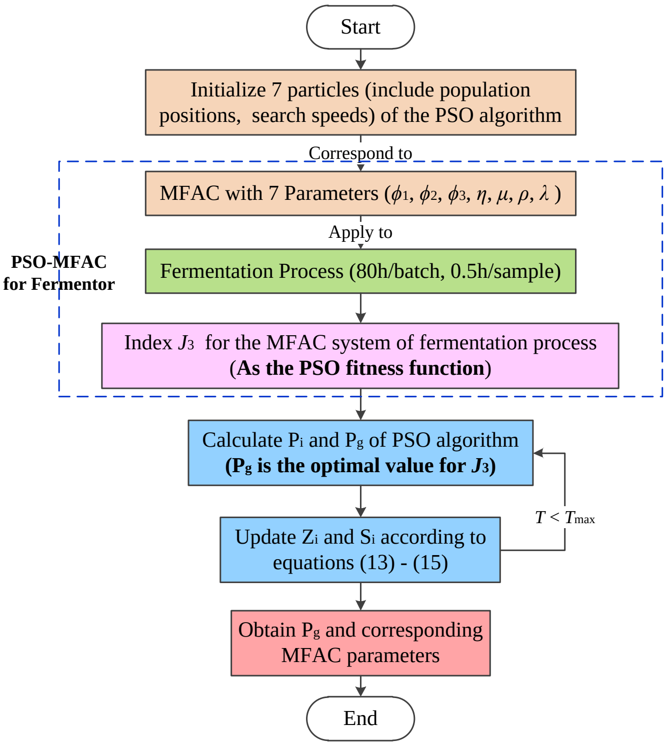

Figure 3. Implementation flowchart of PSO-MFAC control in batch fermentation process

2.4 Batch Fermentation Process

The simulation environment for this article is Matlab R2010a (version 7.10), with a CPU frequency of 2.6 GHz and 8.0 GB of memory. Using the m file of Matlab for programming implementation, the used data can be obtained through the mechanism mathematical model (mathematical differential equation) of the fermentation process. Taking the fermentation process as an example for batch simulation, the fermentation process used is a typical nonlinear process with time-varying characteristics. Studying a simple and efficient fermentation process control method and applying it to a practical biological fermentation experimental set-up can improve the control quality of bacterial mass concentration in the fermentation process, ultimately affecting the product quality and energy consumption in the production process, which is related to the production efficiency of enterprises and has important engineering application value. The following mathematical differential equations can be used to characterize the investigated batch fermentation process to confirm the suggested approach's viability and efficacy. [29-31]:

$$\frac{dX}{dt} = -DX + \mu X \tag{18}$$

$$\frac{dS}{dt} = D(S_f - S) - \frac{\mu X}{Y_{X/S}} \tag{19}$$

$$\frac{dP}{dt} = -DP + (\alpha\mu + \beta)X \tag{20}$$

$$\mu = \frac{\mu_m(1 - P/P_m)S}{K_m + S + S^2/K_{\text{in}}} \tag{21}$$

In the formula, the substrate concentration, product concentration, and flow-added substrate concentration are expressed as S, P, S_f respectively, D is the dilution rate of the fermentation process, and X is the bacterial concentration of the fermentation process. μ is the bacterial growth rate of the fermentation process, which is mainly used to reflect its inhibitory effect on the substrate and the product in the fermentation process. Additionally, expressed in terms of μ_m, P_m, K_m, K_{in} and $Y_{X/S}$ are the highest rate of growth of the bacterium, the coefficient of product saturation, the constant of substrate saturation, the constant of substrate inhibition and the rate coefficient of the bacterium's yield to the substrate during the fermentation process. Nevertheless, α and β represent the fermentation reaction process's kinetic parameters, and specific parameter values and operating conditions can be obtained from the study of Henson and Seborg [29]. The process input variable is usually chosen as the dilution rate (D). The concentration of bacterial (X), the concentration of substrate concentration(S), and the produced substance concentration (P) can all be used as process output variables. It has been shown that if the dilution rate (D) of the fermentation process is reasonably adjusted to control the bacterial concentration (X), to obtain the optimum capacity of production [29], therefore, the concentration of bacterial (X) was chosen as the variable for process output [19].

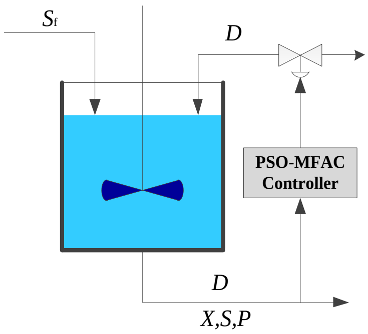

Figure 4. Schematic diagram of batch fermentation process PSO-MFAC control

Figure 4 depicts the batch process control scheme, where controlling X is accomplished by modifying D. The batch fermentation process is decomposed into four subintervals using the control scheme, and PSO-MFAC control is performed 20 hours on the fermentation process with a sub-process time to investigate the affiance of a time-by-time optimal control effect that can be obtained under time-varying process parameters.

3.0 RESULT AND DISCUSSION

3.1 Pseudo-Partial Derivative Parameters of PSO-MFAC

Regarding the batch process control issue, the control performance is improved by obtaining high-quality control parameters through reasonable and effective performance indicators [32]. In order to realize the adaptive control of PSO-MFAC controller for a batch process, firstly, a set of MFAC parameters were selected to be optimized by the PSO algorithm in a certain target range (e.g., desired target concentration of batch fermentation process, X_k^{set} =4.5~5.0), with a batch time of 40 hours, a sampling time of 0.5 hours, and a sampling moment of 0.05 hours, as shown in Table 1. The PSO algorithm was employed to enhance the optimization of the control parameters of MFAC and design the PSO-MFAC controller. Under the condition of concentration X_k^{set} = 5.0 and no parameter time-variation, the tracking control effect for 20 hours is shown in Figure 5. The corresponding online self-adjustment of the PPD parameters of the MFAC controller is shown in Figure 6. The cognitive scaling factor (φ_1) represents the influence of a particle's personal best-known position (p_{best}) on its updated velocity, whereas the social scaling factor (φ_2) represents the influence of the global best-known position (g_{best}) on a particle's updated velocity. These parameters φ_1 and φ_2 define the degree to which a particle updates its position under the influence of its own prior performance (exploration) and the best-known solution in the entire swarm (exploitation) with the value of 0.7416 and -1.1219, respectively. The inertia weight (φ_3) parameter governs the balance between exploration and exploitation [33]. A higher φ_3 value favors exploration, while a lower value favors exploitation. A higher φ_3, often in the range of 0.9 to 1.0 or even higher, encourages more search space exploration. This can be useful to investigate a greater range of control actions to quickly react to significant changes in the system's behavior. Meanwhile, A lower φ_3, commonly ranging from 0.1 to 0.4 or perhaps lower, facilitates the prioritization of exploiting the most optimal options. The utilization of the PSO-MFAC controller can facilitate the convergence towards optimal or near-optimal control actions, particularly when the system dynamics exhibit a certain degree of stability. Therefore, the obtained value of φ_3 was 0.0083, indicating that the φ_3 has a lower value, proves the PSO-MFAC controller favors the exploitation for the optimal option.

Table 1. Parameters of the MFAC controller operating the algorithm of PSO for x_k^{set}=4.5 to 5.5

Parameter	Value
φ_1	0.7416
φ_2	-1.1219
φ_3	0.0083
η	-0.2431
μ_C	0.5042
ρ	0.6396
λ	0.5639

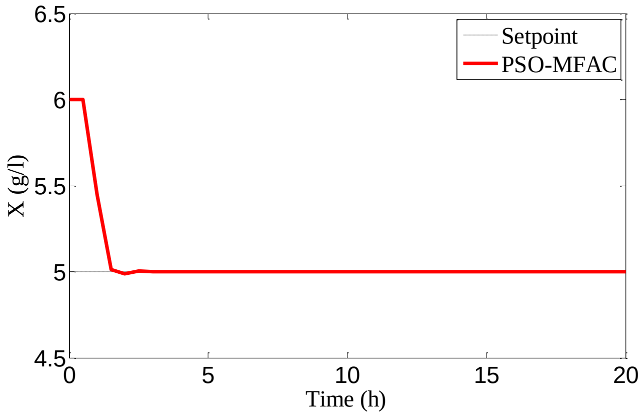

Figure 5. PSO-MFAC set-point of performance tracking at X_k^{set}=5.0

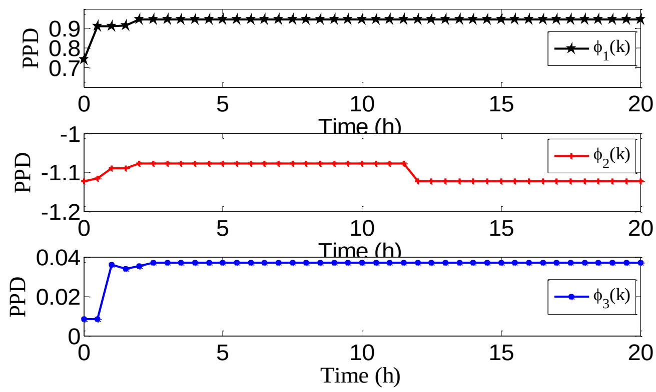

Figure 6. Three Pseudo-partial Derivative Parameters of PSO-MFAC controller for X_k^{set}=5.0

3.2 Fermentation Process Of PSO-MFAC Controller

Batch operations typically have time-varying process parameters and multi-stage sub-interval targets, such as the time-varying, uncertain model parameter problem that is typical of fermentation processes [32]. The batch fermentation is assumed to have a homogeneous mixture, constant density/viscosity, perfect mixing, isothermal conditions, steady initial state, constant reaction kinetics, negligible gas-liquid mass transfer resistance, ideal sensors/actuators, and no external disturbances. Linearization assumes small deviations from equilibrium for approximating nonlinear dynamics. Two crucial factors, namely the maximal growth rate (μ_m) and the coefficient of bacterial yield to substrate ($Y_{X/S}$), are particularly significant to most bio-reaction processes. These parameters are variable and can alter the reaction process's usual course. [32,34]. It has been shown [12,32] that μ_m and $Y_{X/S}$, if they are out of a certain range, the controller is no longer able to regulate them. The time-varying controllable range of μ_m and $Y_{X/S}$ is shown in Table 2.

Table 2. The range of $\mu_m$and $Y_{X/S}$ with x_k^{set} =4.5~7 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
$Y_{X/S}$	0.4~0.6	0.4~0.52	0.4~0.51

In general, control performance is measured in terms of the common ISE metric (Integral Square Error, Integral Square of Error):

$$\mathrm{ISE} = \sum_{k=1}^{\tilde{N}} |e(k)|^2 \tag{22}$$

with the aim of verifying the efficacy of the projected PSO-MFAC control, considering the uncertainty of the time-varying parameters in the above batch fermentation process at μ_m=[0.48,0.65] and $Y_{X/S}$ = [0.3,0.55], the traditional PID [16], simple MFAC [18], and Auto-tuning neural PID (ANPID) [12] were used to compare with the proposed PSO-MFAC method, respectively. As a comparison, the formula structure of the traditional PID control and ANPID method used is as follows, that is:

$$\Delta u(k) = K_p\big(e(k) - e(k-1)\big) + K_i e(k) + K_d\big(e(k) - 2e(k-1) + e(k-2)\big) \tag{23}$$

where, $K_i = K_p t_s T_i^{-1}$ and $K_d = K_p T_d t_s^{-1}$ denote the integral gain coefficient and the differential gain coefficient, respectively.

The assessment of the control performance of the above four methods is presented in Table 3, and PSO-MFAC has a more favorable control effect due to its lowest ISE value, which is 1.1192 compared to the other methods.

Table 3. Comparison of PSO-MFAC control performance and other methods for x_k^{set}=4.5~5.5

Method	Time-varying Range	ISE Value
Traditional PID	μ_m=0.48~0.65,$Y_{X/S}$=0.3~0.55	2.9178
Simple MFAC	μ_m=0.48~0.65,$Y_{X/S}$=0.3~0.55	1.9817
ANPID	μ_m=0.48~0.65,$Y_{X/S}$=0.3~0.55	1.1643
PSO-MFAC	μ_m=0.48~0.65,$Y_{X/S}$=0.3~0.55	1.1192

Figures 7 to 10 represent the fluctuation of the process parameters μ_m and $Y_{X/S}$, depending on the control of process disturbances and noise. In Figure 7, due to the fact that parameter tuning usually relies on experience, conventional PID controllers have very average tracking control effects for nonlinear processes with time-varying parameters. The simple form of MFAC only considers the time-varying dynamic relationship between the output change and input change at a certain moment and has poor stability in the tracking process. As for the adaptive mechanism of the ANPID controller, the PID parameters can be dynamically adjusted according to the time-varying parameters of the nonlinear process, which has a good tracking control effect. The PSO-MFAC adopted fully utilizes the influence of input and output changes in a certain period of time in historical information on the output changes in the next moment. On the other hand, by using the PSO algorithm for intelligent optimization, high-quality control parameters are obtained, thus, the PSO-MFAC tracking curve is closer to the ideal trajectory than the other approaches, i.e., it has the best tracking control effect. Figure 8 depicts the equivalent MFAC pseudo-partial derivative online adaptive scenario, demonstrating that the PSO-MFAC controller may update the control parameters to obtain superior tracking control. Analyzing the cases in Figures 7 and 8, the proposed PSO-MFAC controller is capable of adaptively regulating the batch process based on real-time information to satisfy the state requirements of the corresponding sub-processes as soon as the desired value changes, the step disturbance occurs, and the noise occurs. As the process operation time progresses, the suggested PSO-MFAC control system has a superior time-by-time convergent control effect.

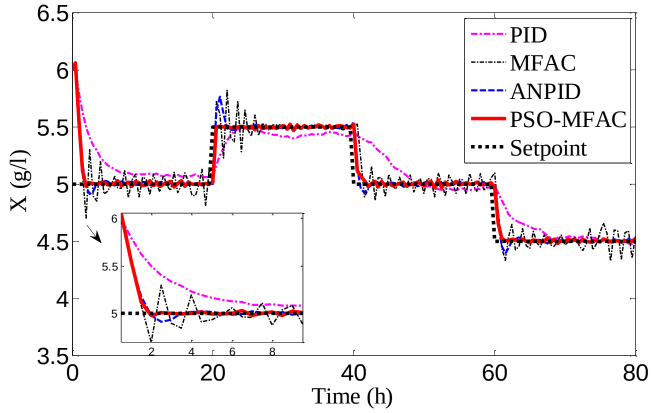

Figure 7. The Comparison of PSO-MFAC Tracking performance and other methods in batch process

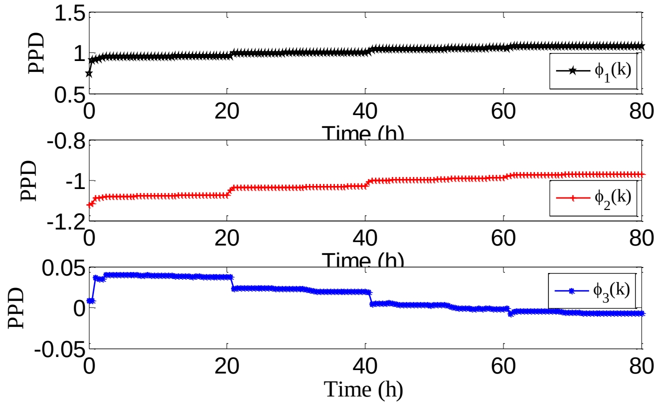

Figure 8. PSO-MFAC control parameters of the batch process

Figure 9 illustrates the time-varying behavior of μ_m against time in batch processing. The time-varying behavior of a batch process's maximum growth rate, particularly in the context of a PSO-MFAC system, refers to how the system's rate of optimizing and adapting to changing conditions develops over time. The μ_m is frequently employed to refer to the rate of convergence or optimization of the PSO algorithm. In the context of PSO-MFAC applied to a batch process, there are several factors that can contribute to the occurrence of time-varying behavior in the maximum growth rate, such as process dynamic, PSO-MFAC parameters, exploitation and exploration, feedback and measurement noise and batch phases. The plot in Figure 9 proves that the growth rate has increased over time since the value of μ_m was 0.5^{-}h at time 0 hour, whereas by the time of 80 hours of batch processing, the value of μ_m has risen to 0.58^{-}h.

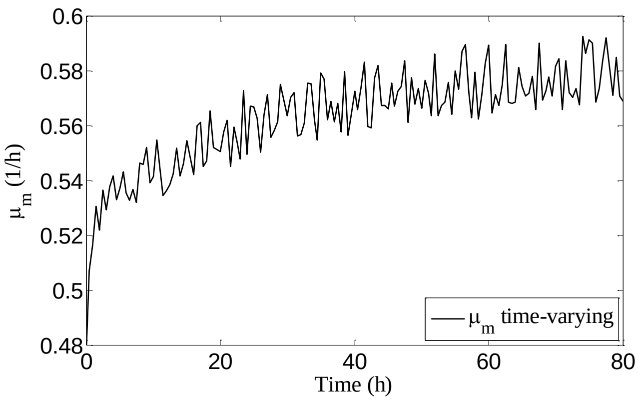

Figure. 9. Time-varying behavior of $\boldsymbol{\mu_m}$ of the batch process

Meanwhile, Figure 10 demonstrates the time-varying behavior towards $Y_{x/s}$. During a batch process, the time-varying behavior of the $Y_{x/s}$ comprises real-time control parameter optimization to ensure that the system adapts to changing circumstances. The plot shows that at 0 hour to 20 hours was the initial state with the value of 0.4^{-}h in range. During the initial stage of the batch process, it is common for the $Y_{x/s}$ to reach its maximum value. This is mostly attributed to the presence of a higher amount of substrate and the bacteria being in a period of exponential growth. The growth phases of the bacteria influence the time-varying behavior of the $Y_{x/s}$. The coefficient may decrease when the bacteria progress from the exponential growth phase to the stationary phase, as shown in the plot at time 60 hours to 80 hours. In addition, considering the control problems of batch processes that may exist in the cases of expectation setting, repetitive operation, and under incomplete process information, further research directions include parameter optimization for data-driven setting tuning MFAC [35], data-driven control in a two-dimensional framework [36], control methods under incomplete information [37], control methods based on active learning [38] dynamic data reconciliation [39], etc.

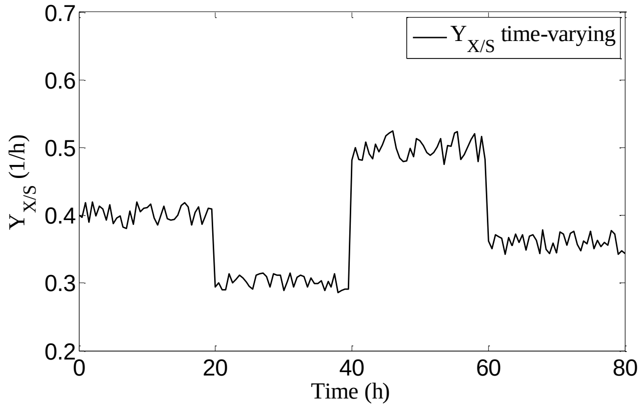

Figure 10. Time-varying behavior of $Y_{X/S}$ for the batch process

4.0 CONCLUSION

Considering a number of processes are difficult to adequately model for control, establishing data-driven control methods for time-varying batch processes is a challenge of major practical significance. The suggested PSO-MFAC control approach is basic in structure and straightforward to implement, and it achieves good tracking control outcomes by updating the main controller parameters for nonlinear and time-varying parameter problems found in batch processes. In summary, compared with the traditional PID control, simple MFAC control and ANPID control methods, the PSO-MFAC control method, with simple structure, is easy to realize and robust. The adoption of the PSO-MFAC method was predicated on its proven efficacy in handling nonlinearities and time-varying parameters in batch processes, as substantiated by prior studies. The proposed PSO-MFAC control scheme is an effective and feasible control method for the nonlinear and time-varying parameter problems of batch processes.

5.0 ACKNOWLEDGEMENTS

This research was supported by the Ministry of Higher Education (MOHE) through the Fundamental Research Grant Scheme (FRGS) (FRGS/1/2023/ICT06/UTHM/02/2).

6.0 DECLARATION OF INTEREST STATEMENT

Ethics approval and consent to participate

Every procedure carried out in this study that involved human subjects complied with the national and/or institutional research committee's ethical guidelines.

Availability of data and materials

The study's data are available inside the publication and/or its additional materials, as confirmed by the authors. The corresponding author* certifies that the study presented in this publication is truthful, accurate, and transparent; no significant study-related errors have been made; and any deviations from the intended study (and, if applicable, recorded) have been explained.

Competing Interests

The authors declare that they have no competing interests.

7.0 REFERENCES

[1] Y. Liu, M. Jia, D. Xu and T. Yang, "Physics-guided graph learning soft sensor for chemical processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 249, p. 105131, 2024.

[2] J. Zhu, M. Jia, Y. Zhang, W. Zhou, H. Deng and Y. Liu, "Domain adaptation graph convolution network for quality inferring of batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 243, no. 15, p. 105028, 2023.

[3] D. Bonvin, G. Francois, G. Rizzi and M. Amrhein, "Chapter 8: Real-time optimization of batch processes via optimizing feedback control," in *Simulation and Optimization in Process Engineering*: *The Benefit of Mathematical Methods in Applications of the Process Industry*, M. Bortz and N. Asprion, Ed., Amsterdam, Netherlands: Elsevier, 2022, pp. 201-223.

[4] Y. Shi, X. Hu, Z. Zhang and L. Xie, "Batch-to-batch optimization for economic performance improvement in batch processes by rational disturbances utilization," *Chemical Engineering Research and Design*, vol. 190, no. 9, pp. 651-666, 2022.

[5] P. Lu, J. Chen and L. Xie, "Iterative learning control (ILC)-Based Economic Optimization for Batch Processes using helpful disturbance information," *Industrial & Engineering Chemistry Research*, vol. 57, no. 10, pp. 3717-3731, 2018.

[6] H. Yoo, H. E. Byun, D. Han and J. H. Lee, "Reinforcement learning for batch process control: Review and perspectives," *Annual Reviews in Control*, vol. 52, pp. 108-119, 2021.

[7] J. Lu, Z. Cao and F. Gao, "Batch process control: overview and outlook," *Acta Automatica Sinica*, vol. 43, no. 6, pp. 933-943, 2017.

[8] E. A. D. Rio-Chanona, P. Petsagkourakis, E. Bradford, J. E. A. Graciano and B. Chachuat, "Real-time optimization meets Bayesian optimization and derivative-free optimization: A tale of modifier adaptation," *Computers & Chemical Engineering*, vol. 147, no. 4, p. 107249, 2021.

[9] D. Rodrigues, A. Marchetti and D. Bonvin, "On improving the efficiency of modifier adaptation via directional information," *Computers & Chemical Engineering*, vol. 164, no. 9, p. 107867, 2022.

[10] J. Wang, J. Zhang, F. Zhang, Z. Sun, Z. Sui and Y. Tian, "Measurement-based two-layer optimization scheme for machining system with repetitive and batch characteristics," *The International Journal of Advanced Manufacturing Technology*, vol. 121, no. 3, pp. 2553-2563, 2022.

[11] L. Ye, "Within-batch self-optimizing control for batch processes," *Acta Automatica Sinica*, vol. 48, no. 11, pp. 2777-2787, 2022.

[12] Z. Wang, C. Chen, X. Chen, D. Li and F. Zeng, "Adaptive PID control for time-varying fermentation processes," presented at IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS), Suzhou, Jiangsu, China, 2021, May 14-16.

[13] X. Yu, Z. Hou and M. M. Polycarpou, "Controller-dynamic-linearization-based data-driven ILC for nonlinear discrete-time systems with RBFNN," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 8, pp. 4981-4992, 2022.

[14] C. Zhou, L. Jia and Y. Zhou, "Tube-based iterative-learning-model predictive control for batch processes using pre-clustered just-in-time learning methodology," *Chemical Engineering Science*, vol. 259, no. 6, p. 117802, 2022.

[15] R.P. Borase, D.K. Maghade, S.Y. Sondkar, S.N. Pawar. A review of PID control, tuning methods and applications, *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818-827, 2020.

[16] Q. Liu, M. Jia, X. Guo, "Two-dimensional LSTM soft sensor using noisy process data," *Measurement Science and Technology*, to be published, 2024.

[17] R. Chi, Z. Hou and B. Huang, "Optimal iterative learning control of batch processes: from model-based to data-driven," *Acta Automatica Sinica*, vol. 43, no. 6, pp. 917-932, 2017.

[18] Z. Pang, B. Ma, W. Song and G. Liu, "An improved compact form model free adaptive control method," *Control and Decision*, vol. 36, no. 2, pp. 436-442, 2021.

[19] Y. Liu, C. Lin, and Z. Gao. "Particle swarm optimization based PID control for continuous fermentation process," *Automation Petro-Chemical Industry*, vol. 48, no. 1, pp. 52-55, 2012.

[20] Z. Hou, S. Liu and C. Yin, "Local learning-based model-free adaptive predictive control for adjustment of oxygen concentration in syngas manufacturing industry," *IET Control Theory and Applications*, vol. 10, no. 12, pp. 1384-1394, 2016.

[21] R. Chi and Z. Hou, "A model free adaptive control approach for freeway traffic density via ramp metering," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 11, pp. 2823-2832, 2008.

[22] S. Di and H. Li. "Model free adaptive speed control on the travelling wave ultrasonic motor," *Journal of Electronical Engineering*, vol. 69, no. 1, pp. 14-23, 2018.

[23] Z. Hou and S. Jin, "Chapter 3: Dynamic linearization approach of discrete-time nonlinear systems," in *Model free adaptive control: Theory and applications*, Taylor and Francis Group, Ed., New York, USA: CRC Press, 2013, pp. 45-71.

[24] Z. Ji, H. Liao and Q. Wu, "Chapter 1: Introduction," in *Particle swarm algorithms and applications*, H. Zhang, Eds, Beijing, China: Science Press, 2009, pp. 1-15.

[25] R. Chi, H. Zhang, H. Li, B. Huang and Z. Hou, "Data-driven dynamic internal model control," *IEEE Transactions on Cybernetics*, to be published, 2024.

[26] J. Zheng and Z. Hou, "ESO-based model-free adaptive iterative learning energy-efficient control for subway train with disturbances and over-speed protection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8136-8148, 2023.

[27] W. Chang, "An improved PSO algorithm for solving nonlinear programing problems with constrained conditions," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 12, no. 1, pp. 2150001-2150016, 2021.

[28] W. Chang, "An improved particle swarm optimization with multiple strategies for PID control system design," *International Journal of Modeling and Optimization*, vol. 12, no. 2, pp. 54-60, 2022.

[29] M. A. Henson and D. E. Seborg, "An internal model control strategy for nonlinear system," *American Institute of Chemical Engineers Journal*, vol. 37, no. 7, pp. 1065-1081, 1991.

[30] T. K. Radhakrishnan, S. Sundarams and M. Chidambaram, "Nonlinear control of continuous bioreactors," *Bioprocess Engineering*, vol. 20, no. 2, pp. 173-178, 1999.

[31] Ch. Venkateswarlu and K. V. S. Naidu, "Dynamic fuzzy model based predictive controller for a biochemical reactor," *Bioprocess Engineering*, vol. 23, no. 2, pp. 113-120, 2000.

[32] Y. Liu, W. Chen, H. Wang, Z. Gao and P. Li, "Adaptive control of nonlinear time-varying processes using selective recursive kernel learning method," *Industrial and Engineering Chemistry Research*, vol. 50, no. 5, pp. 2773-2780, 2011.

[33] S. Zdiri, J. Chrouta and A. Zaafouri, "An expanded heterogeneous particle swarm optimization based on adaptive inertia weight," *Mathematical Problems in Engineering*, vol. 4, pp. 1-24, 2021.

[34] Y. Gao, Z. Zhao and F Liu, "DMFA-based operation model for fermentation processes," *Computers and Chemical Engineering*, vol. 109, no. 4, pp. 138-150, 2018.

[35] N. Lin, R. Chi, Y. Liu, Z. Hou and B. Huang, "Data-driven set-point tuning of model-free adaptive control," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 13, pp. 7667-7686, 2023.

[36] R. Chi, H. Li, D. Shen, Z. Hou and B. Huang, "Enhanced P-type control: Indirect adaptive learning from set-point updates," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 1600-1613, 2023.

[37] D. Shen, "Iterative learning control with incomplete information: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 5 pp. 885-901, 2018.

[38] T. Chen, Y. Liu and J. Chen, "An integrated approach to active model adaptation and on-line dynamic optimization of batch processes," *Journal of Process Control*, vol. 23, no. 10, pp. 1350-1359, 2013.

[39] W. Zhu, Z. Zhang and Y. Liu, "Dynamic data reconciliation for improving the prediction performance of the data-driven model on distributed product outputs," *Industrial and Engineering Chemistry Research*, vol. 61, no. 51, pp. 18780-18794, 2023.