



Process Systems Engineering and Process Safety

Learning control of fermentation process with an improved DHP algorithm[☆]

Dazi Li ^{*}, Ningjia Meng, Tianheng Song

Institute of Automation, Beijing University of Chemical Technology, Beijing 100029, China

ARTICLE INFO

Article history:

Received 23 October 2015

Received in revised form 26 February 2016

Accepted 3 March 2016

Available online 3 July 2016

Keywords:

Dual heuristic programming

Batch process

Ethanol fermentation process

Learning control

ABSTRACT

Control of the fed-batch ethanol fermentation processes to produce maximum product ethanol is one of the key issues in the bioreactor system. However, ethanol fermentation processes exhibit complex behavior and nonlinear dynamics with respect to the cell mass, substrate, feed-rate, etc. An improved dual heuristic programming algorithm based on the least squares temporal difference with gradient correction (LSTDC) algorithm (LSTDC-DHP) is proposed to solve the learning control problem of a fed-batch ethanol fermentation process. As a new algorithm of adaptive critic designs, LSTDC-DHP is used to realize online learning control of chemical dynamical plants, where LSTDC is commonly employed to approximate the value functions. Application of the LSTDC-DHP algorithm to ethanol fermentation process can realize efficient online learning control in continuous spaces. Simulation results demonstrate the effectiveness of LSTDC-DHP, and show that LSTDC-DHP can obtain the near-optimal feed rate trajectory faster than other-based algorithms.

© 2016 The Chemical Industry and Engineering Society of China, and Chemical Industry Press. All rights reserved.

1. Introduction

Ethanol fermentation is a complex chemical reaction process. Properties of noise, nonlinear, and dynamics make it difficult to control. An optimal substrate feeding policy problem was proposed in [1], where the fed-batch fermentation process was described by four inner state variables including the substrate concentration and the product concentration. In [2], the dynamic reoptimization problem of a fed-batch biochemical process was studied to make the applicability of a class of adaptive critic designs (ACDs) for online reoptimization.

As pointed out in [3], the objectives of fed-batch processes contain maximizing biomass concentration, and minimizing ethanol formation. Also in [3], the genetic search approaches are used to solve these problems. Nonlinear optimal control methods developed for the fed-batch fermentation process have been widely researched [4]. However, due to the difficulties in designing nonlinear controllers for chemical processes with uncertain dynamics, intelligent control methods including various learning control techniques have been studied in recent years [5,6].

As studied in [7], a controller based on a dynamic programming method for a fed-batch bioreactor was developed to find the optimal glucose feed rate profile. Based on the Bellman optimality principle, the dynamic programming methods are mainly used to solve the optimization problems of the dynamic processes with the constraint

conditions. However, the original dynamic programming algorithms suffer from the “curse of dimensionality”. Consequently, these methods show no good performance for the complex, nonlinear continuous environments.

Developments of Q-Learning algorithm [8] still cannot avoid the problem of the “curse of dimensionality”. Aiming at this problem, certain progresses have been made. ACDs [9] can approximately obtain the system state value functions through establishing a “critic”, that is, in order to obtain optimal solution, ACDs make use of function approximation structure (e.g., the neural network) to approximate the value functions. As a class of ACDs, heuristic dynamic programming (HDP) is widely studied as an important learning control method for a bioreactor system [10]. As well known, dual heuristic programming (DHP) algorithm needs more model information but has better learning control effect, especially when the system parameters and environmental conditions are time-variant [11,12]. HDP, DHP and Globalized DHP (GDHP) [13] are all belong to adaptive dynamic programming (ADP) [14]. They are applied to the online learning control and optimal control problems [15,16]. In recent years, Xu *et al.* proposed kernel DHP and its performance is analyzed both theoretically and empirically [17,18].

In the critic learning of DHP, the critic network constructed with neural network shows low efficiency in previous research. Furthermore, derivatives of state value functions are affected by the setting of initial weights, structure selection and iteration speed in the neural network. To solve the above problems, a new DHP algorithm integrating the least squares temporal difference with gradient correction (LSTDC) algorithm [19] as the critic is developed in this paper. The main purpose is replacing neural network structure with LSTDC to simplify the weight

[☆] Supported by the National Natural Science Foundation of China (61573052).

^{*} Corresponding author.

E-mail address: lidz@mail.buct.edu.cn (D.Z. Li).

adjustment process and to improve approximation precision and convergence speed. Then the improved DHP algorithm based on LSTDC (LSTDC-DHP) is applied to realize efficient online learning control for the fed-batch ethanol fermentation process. Simulation results demonstrate that LSTDC-DHP can obtain the near-optimal feed rate trajectory in continuous spaces.

The rest of this paper is organized as follows. Following the introduction, the model of fed-batch ethanol fermentation process is described in Section 2. A review of the Markov decision processes (MDPs) and ACDs is presented in Section 3. The proposed LSTDC-DHP is derived in Section 4. To show the applicability of LSTDC-DHP, simulation experiments are implemented and empirical results are presented in Section 5. Finally the main conclusions are summarized in Section 6.

2. Problem Formulation

The studied process is the fed-batch ethanol fermentation process presented in [20]. The main components in reaction tank contain cell mass, nutrients, water and substrate. The most prominent feature of this process plant is continuously fed by concentrated substrate shown in Fig. 1. The chemical reaction is shown in Eq. (1).

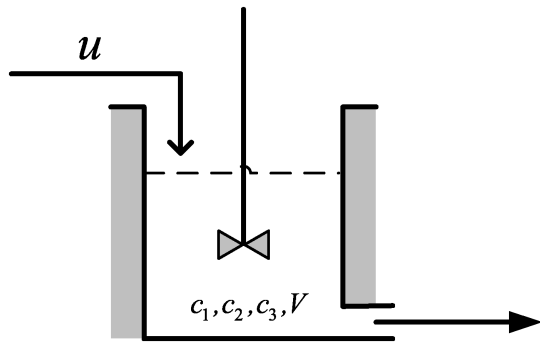
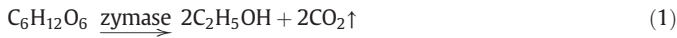


Fig. 1. The ethanol fermentation plant.

In a certain temperature, the raw materials including glucose, zymase and ethanol take complex dynamic biochemical reactions.

The reaction mechanistic model can be described by the following equations:

$$\begin{aligned} \frac{dc_1}{dt} &= \varepsilon c_1 - \frac{c_1}{V} u \\ \frac{dc_2}{dt} &= -10\varepsilon c_1 + \frac{(150-c_2)}{V} u \\ \frac{dc_3}{dt} &= \tau c_1 - \frac{c_3}{V} u \\ \frac{dV}{dt} &= u \end{aligned} \quad (2)$$

with

$$\begin{aligned} \varepsilon &= \frac{0.408c_2}{(1+c_3/16)(0.22+c_2)} \\ \tau &= \frac{c_2}{(1+c_3/71.5)(0.44+c_2)} \end{aligned} \quad (3)$$

where c_1 is the cell mass concentration ($\text{g} \cdot \text{L}^{-1}$), c_2 denotes the substrate concentration ($\text{g} \cdot \text{L}^{-1}$), c_3 stands for the product concentration ($\text{g} \cdot \text{L}^{-1}$), and V is the liquid volume of the reactor (L). The feed-rate to the reactor

is denoted by u , which is constrained by $0\text{L} \cdot \text{h}^{-1} \leq u \leq 12\text{L} \cdot \text{h}^{-1}$. The specific growth rate, ε , and the specific productivity, τ , are functions of c_2 and c_3 .

In Eqs. (2) and (3), the assumed measurable initial state contains four variables which are specified as $[c_1, c_2, c_3, V] = [1, 150, 0, 10]$. The liquid volume V of the reactor is limited by 200 L. The final batch time t_f is fixed as 3 h.

Target of the optimal control is to maximize the yield at the end of a fixed processing time of 3 h. The operated variable is the feed-rate of substrate, u . By combining and integrating the last two differential formulas of Eq. (2), the maximum product output can be obtained as follows:

$$J = \max_u \left[(c_3 V)_{t=t_f} - (c_3 V)_{t=0} = \int_0^{t_f} c_1 \tau V dt \right] \quad (4)$$

Thus, the whole continuous optimal control of the fed-batch ethanol fermentation process can be mathematically formulated as:

$$\min_{u(t)} J(u) = -c_3(t_f)V(t_f) \quad (5)$$

Set the initial condition for the fed-batch ethanol fermentation process as:

$$\begin{aligned} c_3(0) &= 0 & V(0) &= 10 \\ V(t_f) &\leq 200 & 0 \leq u &\leq 12 \end{aligned} \quad (6)$$

Eqs. (2), (3) and (6) form the constraint conditions for Eq. (5).

3. Brief Overview on MDPs and ACDs

3.1. Markov decision processes

Markov chain is a kind of discrete stochastic process with Markov property. This kind of stochastic process has the following characteristics: the next state of the random process is only related to the current state, but not to any other historical states. A MDP is composed of a tuple $\{X, A, P, R\}$, where X is the state space, A is the action space, P is the state transition probabilities, and R is the reward function. When a certain state x_t at time-step t is reached, an action a_t is chosen under a fixed policy π_t . This deterministic stationary policy directly maps states to actions, which is denoted as

$$a_t = \pi_t(\cdot|x_t), \quad t = 0, 1, 2, 3, \dots \quad (7)$$

When the actions a_t satisfy Eq. (7), π is called a Markov policy. The objective of a decision maker is to estimate the optimal policy π^* to satisfy

$$J_{\pi^*} = \max_{\pi} J_{\pi} = \max_{\pi} E^{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \quad (8)$$

where $\gamma \in (0, 1)$ is a discount factor for infinite horizon problems; $E^{\pi}[\cdot]$ stands for the expectation with respect to the policy π and the state transition probabilities.

The state value function $V(x_t)$ is defined as the future expected discounted total reward beginning with x_t when getting through the action of the discount factor:

$$V(x_t) = E \left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \right) \quad (9)$$

For any policy π , the value function is defined as infinite time accumulated expected discounted rewards, namely,

$$V_{\pi}(x) = E_{\pi} \left(\sum_{t=0}^{\infty} \gamma^t r_t | x_0 = x \right) \quad (10)$$

Similarly, the state-action value function $Q^{\pi}(x, a)$ is defined as follows:

$$Q^{\pi}(x, a) = E^{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t | x_0 = x, a_0 = a \right] \quad (11)$$

For an MDP, a deterministic optimal policy can be described as

$$\pi^*(x) = \arg \max_a Q^{\pi^*}(x, a) \quad (12)$$

3.2. Adaptive critic designs

In the last few years, ACDs have been widely studied in the field of reinforcement learning. ACDs mainly consist of three parts: the critic network, the model network and the actor network. The basic idea of ACDs is to construct the approximate function that meets the Bellman optimality principle by adjusting the weights of the critic network. ACDs are applicable in the noisy, nonlinear, and nonstationary environments. DHP is a very important class of ACD methods. DHP uses both value function approximation (VFA) and policy gradient learning strategy algorithm to search for a near-optimal control policy in the continuous spaces.

Generally, the training of ACDs is carried out by the policy iteration process of the dynamic programming. The critic network evaluates the control performance of actor network, which is called policy evaluation process. Meanwhile, the actor network produces the control actions and improves its policy according to the evaluation of the critic network, namely, policy improvement process.

4. The Improved DHP Algorithm Based on LSTD

4.1. Previous work on value function approximation

Since the real-world engineering application problems with large or continuous state spaces exist widely, tabular representations of value functions suffer from the difficulties of great computation and large storage requirements. In order to deal with the generalization of Markov chain learning prediction problem with large and continuous state spaces, and to realize the temporal differences (TD) learning, the TD(λ) algorithm [21] based on VFA has been widely studied in the reinforcement learning and operations research community.

The estimated value function represented by a general linear VFA can be denoted as

$$\bar{V}_t(x_t) = \phi^T(x_t) \mathbf{W}_t \quad (13)$$

where $\phi(x) = (\phi_1(x), \phi_2(x), \dots, \phi_n(x))^T$ is the basis function vector of the state space, $\mathbf{W}_t = (w_1, w_2, \dots, w_n)^T$ is a linear weight vector.

The corresponding incremental TD(λ) weights update rule is

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \alpha_t [r_t + \gamma \phi^T(x_{t+1}) \mathbf{W}_t - \phi^T(x_t) \mathbf{W}_t] \mathbf{Z}_{t+1} \quad (14)$$

where α_t is a learning factor, and the eligibility trace vector $\mathbf{Z}_t(s) = (z_{1t}(s), z_{2t}(s), \dots, z_{nt}(s))^T$ is defined as follows:

$$\mathbf{Z}_{t+1} = \gamma \lambda \mathbf{Z}_t + \phi(x_t) \quad (15)$$

For arbitrary $0 \leq \lambda \leq 1$, the linear TD(λ) algorithm with probability 1 converges to the unique solution of the following equation:

$$E_0[A(X_t)] \mathbf{W}^* - E_0[b(X_t)] = 0 \quad (16)$$

where $X_t = (x_t, x_{t+1}, \mathbf{Z}_{t+1}) (t = 1, 2, \dots)$, and $A(X_t)$ and $b(X_t)$ are expressed as

$$A(X_t) = \mathbf{Z}_t^T [\phi^T(x_t) - \gamma \phi^T(x_{t+1})] \quad (17)$$

$$b(X_t) = \mathbf{Z}_t^T r_t \quad (18)$$

Least squares temporal difference (LSTD) algorithm was proposed to overcome the difficulty of selecting learning factor in linear TD(λ) algorithm, and to improve the efficiency of the data and the convergence speed of the algorithm [22]. In LSTD(0), the performance function is defined as

$$J = \sum_{t=1}^T [r_t - (\phi_t^T - \gamma \phi_{t+1}^T) \mathbf{W}]^2 \quad (19)$$

By employing the instrumental variables approach, the least-squares solution of the above equation is given as

$$\mathbf{W}_{\text{LSTD}(0)} = \left\{ \sum_{t=1}^T [\phi_t (\phi_t - \gamma \phi_{t+1})^T] \right\}^{-1} \left(\sum_{t=1}^T \phi_t r_t \right) \quad (20)$$

where ϕ_t is the instrumental variable which is not with respect to the input and output observation noises.

In order to reduce the computational burden of the matrix inversion and to realize online learning, the LSTD(0) algorithm was further derived to the recursive form RLS-TD(0) [22]. The update rules of RLS-TD(0) take the form of

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mathbf{P}_t \phi_t [r_t - (\phi_t - \gamma \phi_{t+1})^T \mathbf{W}_t] / [1 + (\phi_t - \gamma \phi_{t+1})^T \mathbf{P}_t \phi_t] \quad (21)$$

$$\mathbf{P}_{t+1} = \mathbf{P}_t - \mathbf{P}_t \phi_t (\phi_t - \gamma \phi_{t+1})^T \mathbf{P}_t / [1 + (\phi_t - \gamma \phi_{t+1})^T \mathbf{P}_t \phi_t] \quad (22)$$

In LSTD(0) and RLS-TD(0), the decaying factor is 0. Equivalently, the above two algorithms do not employ the eligibility trace. Accordingly, the RLS-TD(λ) algorithm [23] is considered to further improve the efficiency of TD algorithm. Therefore, Eq. (19) is further modified as

$$J = \left\| \sum_{t=1}^T A(X_t) \mathbf{W} - \sum_{t=1}^T b(X_t) \right\|^2 \quad (23)$$

In LS-TD(λ), the corresponding least-squares estimation of the weight vector is calculated as

$$\mathbf{W}_{\text{LS-TD}(\lambda)} = \mathbf{A}_T^{-1} \mathbf{b}_T = \left[\sum_{t=1}^T A(X_t) \right]^{-1} \left[\sum_{t=1}^T b(X_t) \right] \quad (24)$$

where

$$\mathbf{A}_T = \sum_{t=0}^T A(X_t) = \sum_{t=0}^T \mathbf{Z}_t^T [\phi^T(x_t) - \gamma \phi^T(x_{t+1})] \quad (25)$$

$$\mathbf{b}_T = \sum_{t=0}^T b(X_t) = \sum_{t=0}^T \mathbf{Z}_t^T r_t \quad (26)$$

Based on the above derivation, the weight vector update rules of RLS-TD(λ) have the following form

$$\mathbf{K}_{t+1} = \mathbf{P}_t \mathbf{Z}_t / \left\{ 1 + [\boldsymbol{\phi}^T(x_t) - \gamma \boldsymbol{\phi}^T(x_{t+1})] \mathbf{P}_t \mathbf{Z}_t \right\} \quad (27)$$

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mathbf{K}_{t+1} \left\{ r_t - [\boldsymbol{\phi}^T(x_t) - \gamma \boldsymbol{\phi}^T(x_{t+1})] \mathbf{W}_t \right\} \quad (28)$$

$$\mathbf{P}_{t+1} = \mathbf{P}_t - \mathbf{P}_t \mathbf{Z}_t \left\{ 1 + [\boldsymbol{\phi}^T(x_t) - \gamma \boldsymbol{\phi}^T(x_{t+1})] \mathbf{P}_t \mathbf{Z}_t \right\}^{-1} [\boldsymbol{\phi}^T(x_t) - \gamma \boldsymbol{\phi}^T(x_{t+1})] \mathbf{P}_t \quad (29)$$

where $\mathbf{K}_t$ is a recursive gain matrix and $\mathbf{P}_t$ denotes the variance matrix calculated in the weight updating process.

4.2. LSTD-DHP algorithm

When facing with the more complex control problem, the online reinforcement learning algorithm is needed to further speed up the convergence, and to make the control algorithm more stable. Therefore, another improved algorithm, TD with gradient correction (TDC) [24–27], is derived. The mean-square projected Bellman error performance function employed in TDC is described by

$$J = \left\| \tilde{V} - \Pi T \tilde{V} \right\|_D^2 \quad (30)$$

where T is the Bellman operator and Π is a projection that can project $T\tilde{V}$ back to the VFA function space.

Based on the above performance function, the weight update rules of the TDC algorithm can be obtained

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \alpha_t \delta_t \boldsymbol{\phi}_t - \alpha_t \gamma \boldsymbol{\phi}_{t+1} (\boldsymbol{\phi}_t^T \boldsymbol{\omega}_t) \quad (31)$$

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + \eta_t (\delta_t - \boldsymbol{\phi}_t^T \boldsymbol{\omega}_t) \boldsymbol{\phi}_t \quad (32)$$

where α_t and η_t are the learning factors, and $\delta_t = r_t + \gamma \boldsymbol{\phi}_{t+1}^T \mathbf{W}_t - \boldsymbol{\phi}_t^T \mathbf{W}_t$ is the temporal difference error.

Next, a framework of LSTD is introduced to the weight adjustment in the critic learning of DHP. Based on this framework, we present the idea on how to improve the critic network in DHP. Different from previous works, our work focuses on extending LSTD-DHP to learning control problems in the batch fermentation process. The aim is to obtain better performance compared with previous DHPs with TD algorithm. A general framework of DHP is shown in Fig. 2. The main components include a critic, an actor, a model, and a plant. Model is the system model which is established. And the plant is the system process. The critic is used to approximate the derivatives of value functions. The actor receives the plant's current state x_t and outputs the control a_t .

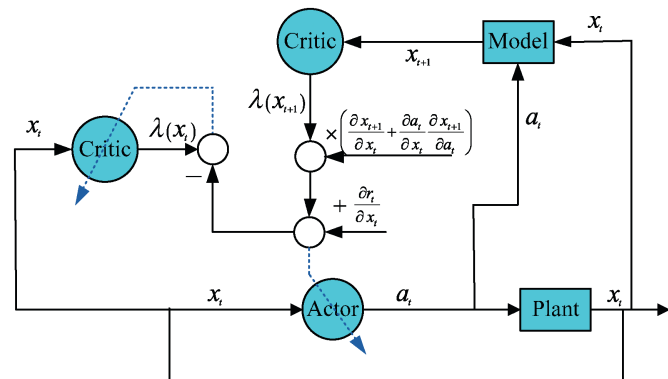


Fig. 2. Learning control structure of DHP.

Then the model receives the control a_t , and estimates the next state x_{t+1} . The polynomials $\frac{\partial x_{t+1}^T}{\partial x_t} + \frac{\partial a_t}{\partial x_t} \frac{\partial x_{t+1}^T}{\partial a_t}$ and $\frac{\partial r_t}{\partial x_t}$ are derived in the next section.

4.2.1. Critic network

ACDs are actor–critic learning control architecture forms. As shown in Fig. 2, the critic network estimates the state value function $V(x_t)$ by using the following Bellman equation of dynamic programming

$$V(x_t) = r_t + \gamma V(x_{t+1}) \quad (33)$$

where r_t is the current reward.

The critic network learning in DHP is to approximate the derivatives of system state value functions:

$$\lambda(x_t) = \frac{\partial V(x_t)}{\partial x_t} \quad (34)$$

where $\lambda(x_t)$ is the output of the critic network.

The goal of critic network is to obtain the mapping relation from x_t to $\lambda(x_t)$. In order to train the critic network, the desired output $\lambda^d(x_t)$ must be computed firstly. According to Eq. (34), $\lambda^d(x_t)$ can be obtained as

$$\lambda^d(x_t) = \frac{\partial V(x_t)}{\partial x_t} = \frac{\partial r_t}{\partial x_t} + \gamma \frac{\partial V(x_{t+1})}{\partial x_t} \quad (35)$$

Further, the derivation of $\lambda^d(x_t)$ can be obtained by using the chain rule iterative

$$\begin{aligned} \lambda^d(x_t) &= \frac{\partial r_t}{\partial x_t} + \gamma \frac{\partial x_{t+1}^T}{\partial x_t} \frac{\partial V(x_{t+1})}{\partial x_{t+1}} \\ &= \frac{\partial r_t}{\partial x_t} + \gamma \left[\frac{\partial x_{t+1}^T}{\partial x_t} + \frac{\partial a_t}{\partial x_t} \frac{\partial x_{t+1}^T}{\partial a_t} \right] \frac{\partial V(x_{t+1})}{\partial x_{t+1}} \end{aligned} \quad (36)$$

Let

$$\frac{\partial V(x_{t+1})}{\partial x_{t+1}} = \lambda(x_{t+1}) \quad (37)$$

where $\lambda(x_{t+1})$ is reflected approximately into the x_{t+1} by the critic network. The derivative $\partial a_t / \partial x_t$ is computed by the backpropagation of the actor network. Meanwhile, $\partial x_{t+1}^T / \partial x_t$ and $\partial x_{t+1}^T / \partial a_t$ can be calculated through the backpropagation of the model network.

The two values $\partial x_{t+1}^T / \partial x_t$ and $\partial x_{t+1}^T / \partial a_t$ in the critic network have very important influence on the result of the iterative convergence, and corresponding computation needs the model information of the control plant. An approximation model obtained by the neural network or system identification methods is enough to provide the model information. In this paper, $\partial x_{t+1}^T / \partial x_t$ and $\partial x_{t+1}^T / \partial a_t$ are referred to as the state Jacobian matrix and action Jacobian matrix in the critic network, respectively.

To approximate the derivatives of state value functions, the main route to improve the DHP algorithm is to change the approximation method of two key parameters $\lambda(x_t)$ and $\lambda(x_{t+1})$. Therefore, VFA is constructed by LSTD which is used to replace the critic neural network structure, namely

$$\lambda(x_t) = \frac{\partial V(x_t)}{\partial x_t} = \boldsymbol{\phi}^T(x_t) \mathbf{W}_t^c \quad (38)$$

$$\lambda(x_{t+1}) = \frac{\partial V(x_{t+1})}{\partial x_{t+1}} = \boldsymbol{\phi}^T(x_{t+1}) \mathbf{W}_{t+1}^c \quad (39)$$

where $\mathbf{W}_t^c$ is the critic weight.

Since the value function parameters make the sum of TD updates over the observed trajectory to be zero, the sums of TDC updates in Eqs. (31) and (32) should be zero as follows:

$$\frac{1}{N} \sum_{i=1}^t (\delta_i - \phi_i^T \omega_t) \phi_i = 0 \quad (40)$$

$$\frac{1}{N} \sum_{i=1}^t (\delta_i \phi_i - \gamma \phi_{i+1} \phi_i^T \omega_t) = 0 \quad (41)$$

Substituting the temporal difference error into Eq. (40), the following equation can be got:

$$\frac{1}{N} \sum_{i=1}^t (r_{i+1} + \gamma \phi_{i+1}^T \mathbf{W}_t^c - \phi_i^T \mathbf{W}_t^c - \phi_i^T \omega_t) \phi_i = 0 \quad (42)$$

Then, Eq. (42) is further modified as

$$\frac{1}{N} \sum_{i=1}^t (\gamma \phi_i \phi_{i+1}^T - \phi_i \phi_i^T) \mathbf{W}_t^c + \frac{1}{N} \sum_{i=1}^t (-\phi_i \phi_i^T) \omega_t = \frac{1}{N} \sum_{i=1}^t (-r_{i+1} \phi_i) \quad (43)$$

By employing the same method for Eq. (41), the following is obtained:

$$\frac{1}{N} \sum_{i=1}^t (r_{i+1} \phi_i + \gamma \phi_i \phi_{i+1}^T \mathbf{W}_t^c - \phi_i \phi_i^T \mathbf{W}_t^c - \gamma \phi_{i+1} \phi_i^T \omega_t) = 0 \quad (44)$$

$$\frac{1}{N} \sum_{i=1}^t (\gamma \phi_i \phi_{i+1}^T - \phi_i \phi_i^T) \mathbf{W}_t^c + \frac{1}{N} \sum_{i=1}^t (-\gamma \phi_{i+1} \phi_i^T) \omega_t = \frac{1}{N} \sum_{i=1}^t (-r_{i+1} \phi_i) \quad (45)$$

The following least-square format for TDC can be obtained by combining Eqs. (43) and (45) together

$$\underbrace{\sum_{i=1}^t \begin{bmatrix} \gamma \phi_i \phi_{i+1}^T - \phi_i \phi_i^T & -\phi_i \phi_i^T \\ \gamma \phi_i \phi_{i+1}^T - \phi_i \phi_i^T & -\gamma \phi_{i+1} \phi_i^T \end{bmatrix}}_{G_t} \underbrace{\begin{bmatrix} \mathbf{W}_t^c \\ \omega_t \end{bmatrix}}_{h_t} = \sum_{i=1}^t \underbrace{\begin{bmatrix} -r_{i+1} \phi_i \\ -r_{i+1} \phi_i \end{bmatrix}}_{h_t} \quad (46)$$

In order to make sure G_t is full rank, the ridge regression is employed by initializing G_t with an identity matrix multiplied by a small positive number g . As a basic ridge regression problem, the effect of g is not only to make G_t invertible, but also to affect the critic weight on regularization. Since few literatures have previously discussed the sensitivity relating to g value choice, the choice of g value is based on trial and error in simulation experiments. The LSTDC update rules for critic learning in LSTDC-DHP are as follows

$$\begin{bmatrix} \mathbf{W}_t^c \\ \omega_t \end{bmatrix} = (G_t + gI)^{-1} h_t \quad (47)$$

$$\begin{cases} G_{t+1} = G_t + \begin{bmatrix} \gamma \phi_t \phi_{t+1}^T - \phi_t \phi_t^T & -\phi_t \phi_t^T \\ \gamma \phi_t \phi_{t+1}^T - \phi_t \phi_t^T & -\gamma \phi_{t+1} \phi_t^T \end{bmatrix} \\ h_{t+1} = h_t + \begin{bmatrix} -r_{t+1} \phi_t \\ -r_{t+1} \phi_t \end{bmatrix} \end{cases} \quad (48)$$

LSTDC is used to construct a proper basis function to replace the critic neural network structure in LSTDC-DHP. Difficulties of selecting parameters in the previous DHP based on neural network function approximation can be overcome. Therefore, the approximation precision and convergence speed of critic network are improved. The critic weight can be rapidly estimated only by constructing a suitable basis function to obtain continuous observation data.

4.2.2. Actor network

The actor network is used to generate the control actions based on the observed states of the plant. The output of the actor network is given by

$$a_t = g(x_t, \mathbf{W}_t^a) \quad (49)$$

where $\mathbf{W}_t^a$ is the actor weight.

The action outputs of the actor are determined by the following Gaussian probabilistic distribution:

$$p_r(u_t) = \exp \left[-\frac{(u_t - a_t)^2}{\sigma_t^2} \right] \quad (50)$$

where u_t is the actual control output and the variance σ_t is defined as follows:

$$\sigma_t = c_1 / \{1 + \exp[c_2 \lambda(x_t)]\} \quad (51)$$

where c_1 and c_2 are positive constants.

In LSTDC-DHP, based on the outputs of the critic, the following policy gradient methods can be used to train the actor network.

$$\begin{aligned} \mathbf{W}_{t+1}^a &= \mathbf{W}_t^a + \beta_t \frac{\partial V(x_t)}{\partial \mathbf{W}_t^a} = \mathbf{W}_t^a + \beta_t \frac{\partial V(x_t)}{\partial a_t} \frac{\partial a_t}{\partial \mathbf{W}_t^a} \\ &\approx \mathbf{W}_t^a + \beta_t \delta_t \frac{u_t - a_t}{\sigma_t} \frac{\partial a_t}{\partial \mathbf{W}_t^a} \end{aligned} \quad (52)$$

4.2.3. Model network

The model network is to approximate the characteristics of dynamic process. The mechanistic model of fed-batch ethanol fermentation process is regarded as the model network.

Algorithm 1. Learning control of fermentation process based on LSTDC-DHP

- (1) Give the initial critic weights $\mathbf{W}_{i0}^c$ and actor weights $\mathbf{W}_{i0}^a$;
- (2) Set $t = 0$ and $J_0 = 0$
- (3) According to the current state x_t , compute the output of the actor network $a_t = g(x_t, \mathbf{W}_t^a)$, select the action u_t , and observe the next state x_{t+1} , get next reward r_{t+1} ;
- (4) Compute the performance index function J_t according to Eq. (4);
- (5) Apply LSTDC described in Eqs. (47) and (48) to update the weights of the critic network;
- (6) Update the weights of the actor network using Eq. (52);
- (7) Until $J_{t+1} - J_t \leq 0$ or the fermentation time reaches t_f ;
- (8) Let $t = t + 1$, return to (3).

In the weight adjustment process of LSTDC-DHP, the weight is estimated by using continuous observation data which is generally easy to obtain. Therefore, LSTDC-DHP can be applied in the batch process with greatly improved efficiency of the weight update and reduced parameter setting. The procedure of LSTDC-DHP is briefly summarized in Algorithm 1.

5. Simulation Studies and Discussion

The learning control of the fed-batch ethanol fermentation process is studied to illustrate the effectiveness of LSTDC-DHP applied to the continuous process. To solve the reinforcement learning problem with continuous state space, a class of linear function approximators, which is called Cerebellar Model Articulation Controller (CMAC) is used. It is well known that CMAC has been widely used in process control and function approximation. In the LSTDC-DHP learning algorithm, a CMAC neural network with four inputs and one output is used in the actor network.

In the experiment, the actor's learning factor is $\beta_t = 0.3$. The weights of the critic network and the actor network are all initialized as 0. G_t is

initialized to an identity matrix and h_t is initialized as 0. The other parameters are set as $\gamma=0.95$, $c_1=0.4$ and $c_2=0.5$. The initial state vector $[c_1, c_2, c_3, V]$ is initialized as $[1, 150, 0, 10]$. The time step is 5 min in the simulation process. A learning control process of ethanol fermenter is defined as starting from the initial state to 36 time steps. The learning control results are shown in Fig. 3. The optimal control policy curve is continuous. The cell mass concentration, the substrate concentration and the product concentration reach steady states at the end of fixed batch time. We can find that the yield of the ethanol fermentation process is becoming smooth gradually in the learning control process. It can also be seen that the performance index converges to the steady-state value eventually.

Considering productive benefit, the specific growth rate and the specific productivity which is in the reaction mechanistic model should be strictly increasing until reaction termination. According to Eq. (3), the product concentration shown in Fig. 3 should decline when the substrate concentration rises for the sake of ascending ε and τ . The liquid volume of the reactor increases all the time. Though the cell mass concentration descends, the cell mass amount is increased on the whole. In order to improve the production, the cell mass needs to consume more substrates. Therefore, the feed-rate to the reactor suddenly rises.

In addition, two other DHP methods integrated with RLSTD(λ) and TDC as the critic algorithms are also implemented for the same control problem, named RLSTD(λ)-DHP and TDC-DHP. Including RLSTD(0)-DHP by Xu *et al.* [18], three methods are used for comparison with LSTDC-DHP. For TDC-DHP, the critic's learning factor is $\alpha_c=0.1$ and η_c is initialized as 0.1. Then, in a similar way, the variance matrix P_t is set to be identity matrix in RLSTD(0)-DHP. λ is equal to 0.6 and the eligibility trace vector is initialized as 0 in RLSTD(λ)-DHP. The performances of the LSTDC-DHP, RLSTD(0)-DHP, RLSTD(λ)-DHP and TDC-DHP algorithms are compared in Fig. 4. The yield of the reactor is chosen as the performance index in the simulation experiments. In LSTDC-DHP, we can see that better yield of the ethanol fermentation process and better values of performance index are obtained as shown in Table 1 due to the use of LSTDC. Fig. 5 shows the product concentration comparison by different algorithms and Fig. 6 shows the control policy curves of the ethanol fermentation process. We can see that the learning

control of the fed-batch ethanol fermentation process is completed continuously. Therefore, suitable feed rate obtained by LSTDC-DHP can lead to the maximum yield.

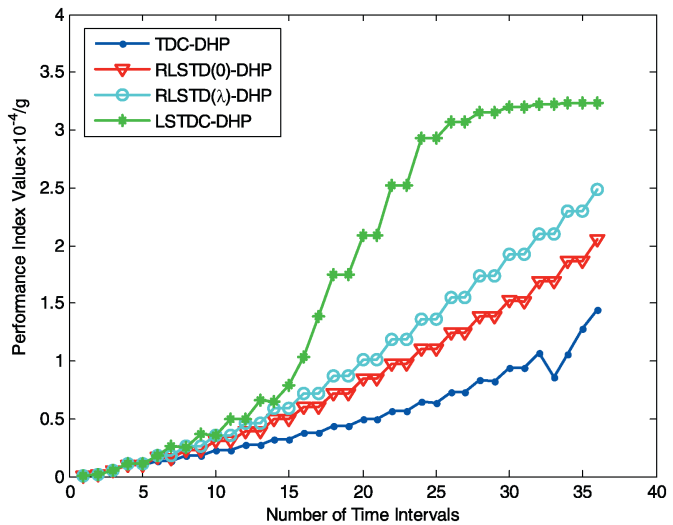


Fig. 4. Performance indexes of four different algorithms.

Table 1
Performance comparison under different algorithms

	TDC-DHP	RLSTD(0)-DHP	RLSTD(λ)-DHP	LSTDC-DHP
Actual performance index	1.4482×10^4	2.0484×10^4	2.4846×10^4	3.2334×10^4

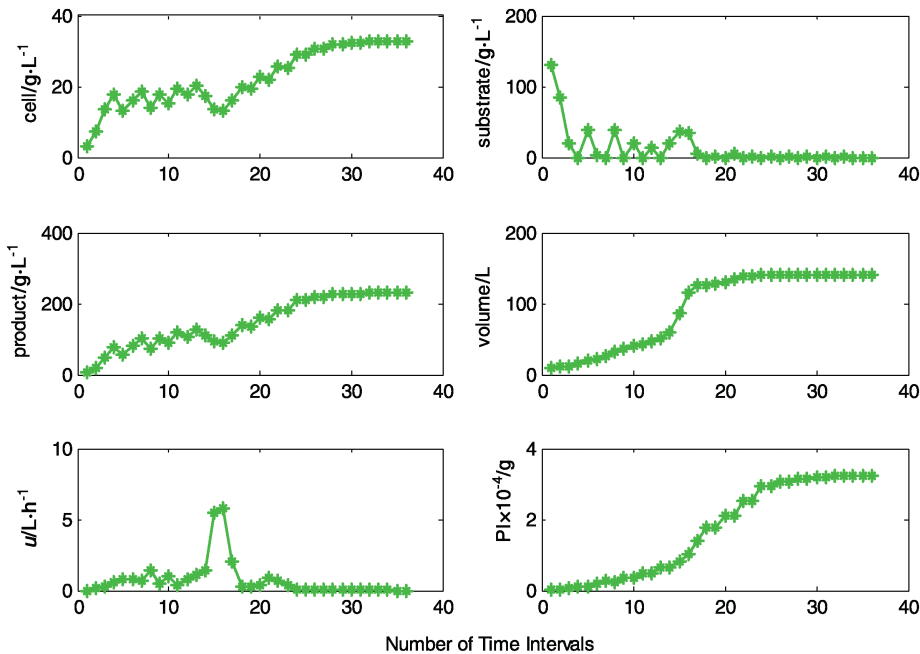


Fig. 3. The learning control results of the fed-batch ethanol fermentation process.

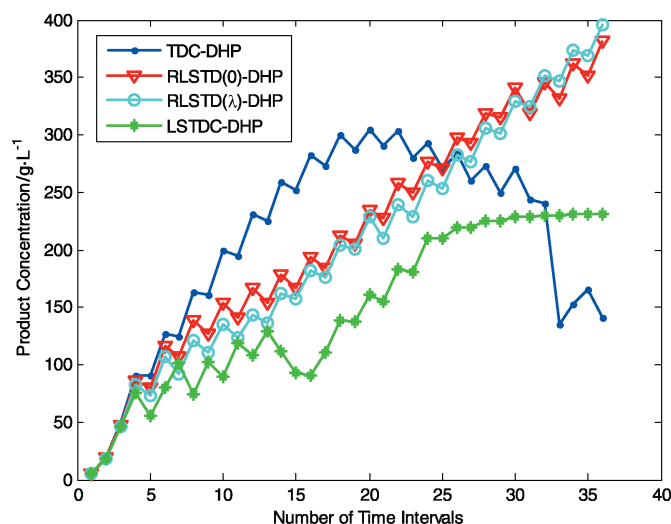


Fig. 5. Comparison of the product concentration by different algorithms.

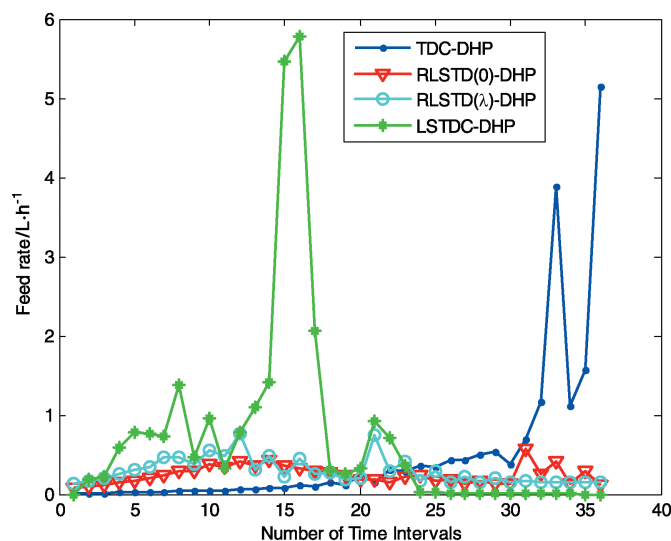


Fig. 6. Curves of control policies by four different algorithms.

6. Conclusions

In this paper, LSTDC-DHP has been developed to deal with learning control problems. LSTDC is chosen as the critic to replace the neural network structure in LSTDC-DHP. Based on the simulations, performance of LSTDC-DHP is analyzed. From the experiment results, LSTDC-DHP takes effect in simplifying the weight adjustment process and improving the critic's approximation precision. LSTDC-DHP can be used to design the optimal feed rate trajectory curve continuously. Consequently, the aim of obtaining the maximum ethanol product is achieved. Simulation results on the learning control of the fed-batch ethanol fermentation process illustrate the effectiveness of LSTDC-DHP and verify the excellent performance of LSTDC-DHP in continuous spaces. This research also shows that it is promising to study the learning control LSTDC-

DHP algorithm and apply it to complex, nonlinear dynamic chemical industrial process efficiently.

References

- [1] J. Hong, Optimal substrate feeding policy for a fed batch fermentation with substrate and product inhibition kinetics, *Biotechnol. Bioeng.* 28 (9) (1986) 1421–1431.
- [2] M.S. Iyer, D.C. Wunsch, Dynamic re-optimization of a fed-batch fermentor using adaptive critic designs, *IEEE Trans. Neural Netw.* 12 (6) (2001) 1433–1444.
- [3] U. Yüzgeç, M. Türker, A. Hocalar, On-line evolutionary optimization of an industrial fed-batch yeast fermentation process, *ISA Trans.* 48 (1) (2009) 79–92.
- [4] A. Ashoori, B. Moshiri, A. Khaki-Sedigh, M.R. Bakhtiari, Optimal control of a non-linear fed-batch fermentation process using model predictive approach, *J. Process Control* 19 (7) (2009) 1162–1173.
- [5] C.V. Peroni, N.S. Kaisare, J.H. Lee, Optimal control of a fed-batch bioreactor using simulation-based approximate dynamic programming, *IEEE Trans. Control Syst. Technol.* 13 (5) (2005) 786–790.
- [6] S. Syafie, F. Tadeo, M. Villafin, A. Alonso, Learning control for batch thermal sterilization of canned foods, *ISA Trans.* 50 (1) (2011) 82–90.
- [7] C. Valencia, G. Espinosa, J. Giral, F. Giral, Optimization of invertase production in a fed-batch bioreactor using simulation based dynamic programming coupled with a neural classifier, *Comput. Chem. Eng.* 31 (9) (2007) 1131–1140.
- [8] D.Z. Li, L. Qian, Q.B. Jin, T.W. Tan, Reinforcement learning control with adaptive gain for a *Saccharomyces cerevisiae* fermentation process, *Appl. Soft Comput. J.* 11 (8) (2011) 4488–4495.
- [9] D.V. Prokhorov, D.C. Wunsch, Adaptive critic designs, *IEEE Trans. Neural Netw.* 8 (5) (1997) 997–1007.
- [10] C.Q. Lian, X. Xu, L. Zuo, Learning control of a bioreactor system using kernel-based heuristic dynamic programming, *Proc. World Congr. Intelligent Control Autom. WCICA 2012*, pp. 316–321.
- [11] B. Wang, D.B. Zhao, C. Alippi, D.R. Liu, Dual heuristic dynamic programming for nonlinear discrete-time uncertain systems with state delay, *Neurocomputing* 134 (2014) 222–229.
- [12] C.Q. Lian, X. Xu, L. Zuo, Z.H. Huang, Adaptive critic design with graph Laplacian for online learning control of nonlinear systems, *Int. J. Adapt. Control Signal Process.* 28 (2014) 290–304.
- [13] M. Fairbank, E. Alonso, D. Prokhorov, Simple and fast calculation of the second-order gradients for globalized dual heuristic dynamic programming in neural networks, *IEEE Trans. Neural Netw. Learn. Syst.* 23 (10) (2012) 1671–1676.
- [14] J. Fu, H.B. He, X.M. Zhou, Adaptive learning and control for MIMO system based on adaptive dynamic programming, *IEEE Trans. Neural Netw.* 22 (7) (2011) 1133–1148.
- [15] Z. Ni, H.B. He, J.Y. Wen, Adaptive learning in tracking control based on the dual critic network design, *IEEE Trans. Neural Netw. Learn. Syst.* 24 (6) (2013) 913–928.
- [16] F.X. Tan, D.R. Liu, X.P. Guan, Online optimal control for VTOL aircraft system based on DHP algorithm, *Proc. of the 33rd Chinese Control Conf., CCC 2014*, pp. 2882–2886.
- [17] X. Xu, Z.S. Hou, C.Q. Lian, H.B. He, Online learning control using adaptive critic designs with sparse kernel machines, *IEEE Trans. Neural Netw. Learn. Syst.* 24 (5) (2013) 762–775.
- [18] X. Xu, C.Q. Lian, L. Zuo, H.B. He, Kernel-based approximate dynamic programming for real-time online learning control: An experimental study, *IEEE Trans. Control Syst. Technol.* 22 (1) (2014) 146–156.
- [19] T.H. Song, D.Z. Li, L.L. Cao, K. Hirasawa, Kernel-based least squares temporal difference with gradient correction, *IEEE Trans. Neural Netw. Learn. Syst.* 27 (4) (2016) 771–782.
- [20] Z.H. Xiong, J. Zhang, Neural network model-based on-line re-optimisation control of fed-batch processes using a modified iterative dynamic programming algorithm, *Chem. Eng. Process.* 44 (4) (2005) 477–484.
- [21] R.S. Sutton, Learning to predict by the method of temporal differences, *Mach. Learn.* 3 (1998) 9–44.
- [22] S.J. Bradtko, A.G. Barto, Linear least-squares algorithms for temporal difference learning, *Mach. Learn.* 22 (1–3) (1996) 33–57.
- [23] X. Xu, H.G. He, D.W. Hu, Efficient reinforcement learning using recursive least-squares methods, *J. Artif. Intell. Res.* 16 (2002) 259–292.
- [24] S. Bhatnagar, D. Precup, D. Silver, Convergent temporal-difference learning with arbitrary smooth function approximation, *Adv. Neural Inf. Process. Syst.-Proc. Conf* 2009, pp. 1204–1212.
- [25] R.S. Sutton, H.R. Maei, D. Precup, Fast gradient-descent methods for temporal-difference learning with linear function approximation, *Proc. Int. Conf. Mach. Learn., ICML 2009*, pp. 993–1000.
- [26] M. Geist, O. Pietquin, Algorithmic survey of parametric value function approximation, *IEEE Trans. Neural Netw. Learn. Syst.* 24 (6) (2013) 845–867.
- [27] H.R. Maei, C. Szepesvári, S. Bhatnagar, R.S. Sutton, Toward off-policy learning control with function approximation, *Proc. Int. Conf. Mach. Learn., ICML 2010*, pp. 719–726.