Data-Driven Flotation Industrial Process Operational Optimal Control Based on Reinforcement Learning

Yi Jiang, *Student Member, IEEE,* Jialu Fan, *Member, IEEE,* Tianyou Chai, *Fellow, IEEE,* Jinna Li, *Member, IEEE,* and Frank L. Lewis, *Fellow, IEEE*

***Abstract*—This paper studies the operational optimal control problem for the industrial flotation process, a key component in the mineral processing concentrator line. A new model-free data-driven method is developed here for real-time solution of this problem. A novel formulation is given for optimal selection of the process control inputs that guarantees optimal tracking of the operational indices while maintaining the inputs within specified bounds. Proper tracking of prescribed operational indices, namely concentrate grade and tail grade, is essential in the proper economic operation of the flotation process. The difficulty in establishing an accurate mathematic model is overcome, and optimal controls are learned online in real time, using a novel form of reinforcement learning we call Interleaved Learning for online computation of the operational optimal control solution. Simulation experiments are provided to verify the effectiveness of the proposed Interleaved Learning method and to show that it performs significantly better than standard Policy Iteration and Value Iteration.**

***Index Terms*—Flotation process; operational optimal control (OOC); model-free; reinforcement learning (RL); interleaved learning.**

I. Introduction

In industrial process control, a common objective is setpoint tracking, where the process outputs, such as flow control, speed control, valve position, etc. track prescribed setpoints while ensuring stability of the closed-loop process. However, for modern complex industrial process, resource usage and economy are becoming more and more important. Therefore, the control goal is not only setpoint tracking of the outputs, but also to achieve good industrial operational performance indices, such as product quality, production efficiency, and production energy consumption. This is known as the operational optimal control (OOC) problem. When the controlled operational indices deviate from the ideal values, the whole control system performance may deteriorate and even result in poor product quality, possibly leading to system shutdown. Therefore, operational control affects not only the quality and efficiency of industrial processes, but also safe and stable operations [1], [2].

One of the most essential procedures in mineral processing is the concentrator process, where crude ore is converted to a highly concentrated useful product. An essential component of the concentrator line is the flotation process, the efficient control of which is the topic of this paper. For the flotation industrial process, the control performance determines the concentrate grade and the metal recovery rate of the concentrator. Thus, the goal of the flotation industrial process operational control is not only to track the basic process outputs, but also to achieve prescribed operational performance indices, namely the concentrate grade and the tail grade.

In the flotation process, the operational performance indices are the concentrate grade and the tail grade, which are related to the process variables, such as feed flow, pulp level, feed air flow, particle size and dosage. Flotation process controllers developed in recent years can automatically control the outputs such as feed flow, pulp level, feed air flow and dosage [3]. In current practice, the setpoints of these variables are prescribed by human operators using their experience to maintain the concentrate grade and the tail grade indices in their target ranges. Nevertheless, when the operating conditions change frequently, the operator cannot prescribe the setpoints accurately and quickly, which leads to wide fluctuations in the concentrate grade and the tail grade, and hence to possible product deteriorations.

In order to overcome the drawbacks of selecting process setpoints manually based on human knowledge and experience, several automatic feedback control approaches are introduced [1]. Researchers have proposed many methods which can generate the output setpoints of industrial processes automatically rather than relying on setpoints prescribed by technical engineers. For some industrial processes, the mechanisms are clear, operate stably, accurate dynamical models can be established. Therefore, researcher engineers have proposed many model-driven control methods to achieve operational performance index control. Model-based methods include as real-time optimization (RTO) [4], [5] and model predictive control (MPC) [6], [7]. Reference [8] combined RTO and

Manuscript received April 21, 2017; revised July 10 and September 4, 2017; accepted September 23, 2017.

This research was supported in part by NSFC 61333012, 61533015, 61304028, 61673280, 111 Project B08015 and the Fundamental Research Funds for the Central Universities N160804001.

Y. Jiang, J. Fan, T. Chai and J. Li are with the State Key Laboratory of Synthetical Automation for Process Industries and International Joint Research Laboratory of Integrated Automation, Northeastern University, Shenyang 110819, China, and Yi Jiang is a Visiting Scholar with the UTA Research Institute, the University of Texas at Arlington, Texas 76118, USA. (e-mail: JY369356904@163.com, jlfan@mail.neu.edu.cn, tychai@mail.neu.edu.cn, lijinna@syuct.edu.cn. J. Fan is the corresponding author with phone +86-24-8368-3410 and fax +86-24-2389-5647)

J. Li is with the School of Information Engineering, Shenyang University of Chemical Technology, Liaoning 110142, P.R. China. (lijinna@syuct.edu.cn)

F.L. Lewis is with the UTA Research Institute, the University of Texas at Arlington, Texas 76118, USA and is also a Qian Ren Consulting Professor, the State Key Laboratory of Synthetical Automation for Process Industries and International Joint Research Laboratory of Integrated Automation, Northeastern University, Shenyang 110819, P.R. China. (lewis@uta.edu)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier: 10.1109/TII.2017.2761852

feedback control, proposing a dynamic compensation setpoint method to make the flotation process operate in good economic conditions. Papers [9], [10] use MPC to optimize the process economics to make the flotation process operate with good economic outcomes. Paper [11] combined RTO and MPC for proper economic control of the flotation process.

Unfortunately, RTO uses steady-state models and MPC relies on an accurate process dynamics model. For some flotation processes, various dynamic mechanisms are not clear and some parameters of these mechanism dynamics are hard to measure online, so that the required flotation process models are hard to establish. Therefore, such model-driven control methods often cannot apply to practical control of flotation processes.

To confront this problem, some researchers proposed data-driven control methods to develop controllers, without knowing dynamical models of the system, to achieve operational performance index control [12]. Paper [13] proposed an intelligence-based supervisory optimal operational controller for a grinding system. Paper [14] proposed a data-driven optimization solution for operational index control that selects the setpoints of the controlled process variables for a grinding system; the solution does not require all the grinding system dynamics. Paper [15] proposed a data-driven optimization control for safe operation of a hematite grinding process. The optimal solution does not require the grinding system dynamics, but this method ignores the physical constraints of the industrial process. Paper [16] proposed a computationally efficient framework for intelligent critic control design and applied it to a smart microgrid. Paper [17] proposed an improved the critic learning criterion to cope with the event-based nonlinear $H\infty$ state feedback control design. In paper [18] the infinite-horizon robust optimal control problem for a class of continuous-time uncertain nonlinear systems is investigated by using data-based adaptive critic designs.

This paper develops a model-free data-driven method for operational optimal control for a single-cell flotation industrial process, which is based on reinforcement learning (RL) [19]–[27]. This method includes three neural networks (NN) [28], [29], one critic NN is used to approximate a certain performance value function, a second actor NN is used to generate the optimal control, while a third model NN is employed to identify the nonlinear process dynamics. Paper [30] studies a Value Iteration (VI) algorithm to identify the nonlinear plant dynamics. That paper uses iterations between a single VI step and a single control update. Our paper, presents a new algorithm which is intermediate between VI and Policy Iteration (PI). Specifically, it iterates between a single PI step and a single control update. The proof of convergence turns to be simpler than the proof in [30] for VI.

The contributions of this paper are as follows. A new dynamical performance index is introduced that allows solution of the OOC problem while enforcing constraints on the process control inputs. It is formally proven that the resulting ideal controller based on the Hamiltonian provides optimality and stability of the closed-loop process. A new Interleaved Learning Algorithm is given where, in contrast to standard PI, the critic NN and actor NN are each updated once in each iteration. A formal proof of convergence of Interleaved Learning is given. Simulation experiments are given that show the performance of Interleaved Learning is better than PI, which converges slowly because the critic NN and actor NN are each tuned to convergence in each iteration.

The paper is organized as follows. In Section II we discuss the Flotation Process and its dynamics. We formulate the flotation cell operational optimal control problem. In Section III we solve the operational optimal control problem, giving the ideal solution. In Section IV, a new algorithm that uses Interleaved Learning to find the optimal solution online. In Section V we provide a simulation experiment to verify the performance of the new Interleaved Learning algorithm, and show further that it performs significantly better than standard Policy Iteration and Value Iteration techniques.

II. Flotation Industrial Process Operational Control Description and Control

One of the most essential procedures in mineral processing is the concentrator process, where crude ore is converted to a highly concentrated useful product. An essential component of the concentrator line is the flotation process, the efficient control of which is the topic of this paper. Here we describe the flotation process and formulate the problem of operational optimal control.

A. Single-Cell Flotation Industrial Process Description

The concentrator flow line is illustrated in Fig. 1, and includes the crush process, the grinding process [13]–[15], the thickener process [31], [32], and the flotation process. The crude ore is first broken through the crush process, then the broken ore is mixed with water and sent to the grinding machine, where the ore becomes ore pulp. The concentration of the ore pulp is increased by the thickener process. Finally, the ore pulp is separated into concentrate and tail by the flotation process; the concentrate is used for smelting, while the tail will be sent to the tailings dam to be discarded.

The flotation process is shown in Fig. 1 and is composed of multiple cells. This paper focuses on control of a single cell, as depicted schematically in Fig. 3. The single-cell flotation industrial process includes the feed pump, flotation cell, concentrate pump, tail pump, and stirring paddle. In operation, the feed ore pulp produced by the thickening process is fed to the flotation cell by the feed pump, then it is agitated by the stirring paddle in the flotation machine, or by ventilating in a ventilation type flotation machine. The feed ore pulp is thereby separated into concentrate pulp, which is the desirable portion, and tail pulp which is discarded. Finally, the concentrate pulp and the tail pulp flow out of the flotation cell through the concentrate pump and tail pump, respectively.

The performance of industrial processes such as flotation is measured by *operational indices* which quantify product quality, resource usage, makespan, and so on. In the flotation industrial process, the key operational indices include the concentrate grade $y_1(k)$ and the tail grade $y_2(k)$, which are influenced by the control inputs and other parameters, and k is the sampling time. The control inputs of the single-cell

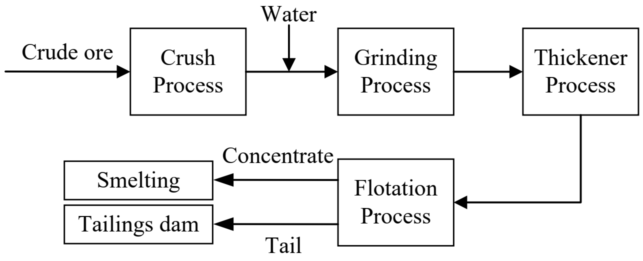


Fig. 1. Concentrator flow line.

Fig. 2. Flotation industrial process.

flotation cell are the pulp height $u_1(k)$ and the feed flow $u_2(k)$. Other measurable parameters influencing the outputs are the ventilation volume $c_1(k)$ and the dosage $c_2(k)$. In addition to these control inputs and measurable parameters, the concentrate grade and the tail grade are influenced by various unmeasurable parameters, such as the size of the mineral particles $d_1(k)$ and the grade of the feed ore $d_2(k)$.

Based on this description, the concentrate grade and the tail grade can be described by the output vector $y(k) = [y_1(k), y_2(k)]^T \in R^2$ at time k, the control input by $u(k) = [u_1(k), u_2(k)]^T \in R^2$, the measurable parameters by $c(k) = [c_1(k), c_2(k)]^T \in R^2$, and the unmeasurable parameters by $d(k) = [d_1(k), d_2(k)]^T \in R^2$. In industrial application, the collector current and frequency of the dosing pump are generally fixed at constant values so that the ventilation volume $c_1(k)$ and the dosage $c_2(k)$ are constants, which are set by a human operator (Technical Department Personnel) using his previous experience and process knowledge. The size of the mineral particles $d_1(k)$ is controlled by gridding system in the grinding process, which is also constant. The grade of the feed ore $d_2(k)$ is determined by the crude ore. Usually, the crude ore is from the same producer, thus, $d_2(k)$ is likewise constant.

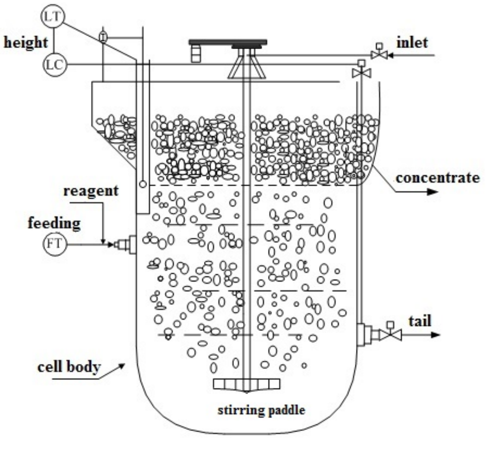


Fig. 3. Schematic illustration of single-cell flotation industrial process.

The $c(k)$ and $d(k)$ will influence the flotation rate, which are the key parameters of the flotation process, the relationship is hard to be modeled. The $c(k)$ is usually set by a human operator (Technical Department Personnel) using his previous experience and process knowledge, and $d(k)$ is determined by the Grinding process and the crude ore as above. In sum, the dynamics of the single-cell flotation process are described by the nonlinear state-space model

$$y(k+1) = f(y(k), u(k), c(k), d(k)) = F(y(k), u(k)) \quad (1)$$

In this equation, $f(\cdot)$ does not depend on $c(k)$, $d(k)$, whereas nonlinear function $F(\cdot)$ does depend on $c(k)$, $d(k)$. Since these unknown parameters are constant, $F(\cdot)$ is unknown and the definition of $F(\cdot)$ based on $f(\cdot)$ is straightforward.

B. Single-Cell Flotation Industrial Process Operational Optimal Control Problem

The current method of controlling the flotation cell relies on human operators and is as follows. In the concentrator, the installed flotation cell control system automatically controls the pulp height $u_1(k)$ and the feed flow $u_2(k)$ to conform to prescribed setpoints by using the feed pump, concentrate pump and tail pump. A human operator (Technical Department Personnel) determines the setpoints for pulp height and feed flow using his previous experience and process knowledge so that the concentrate grade and the tail grade are maintained within prescribed target ranges. Unfortunately, when the operating conditions change frequently, the operator cannot accurately identify the parameters of the flotation process and hence cannot properly determine suitable setpoints accurately and quickly. This leads to the concentrate grade and the tail grade fluctuating in wide-ranges that may fall outside the target ranges. Furthermore, it is virtually impossible for human operators to determine *optimal* setpoints that can yield the best possible values of the operational indices.

To correct these deficiencies, it is desired in this paper to formalize the flotation cell control problem by automatically determining the optimal setpoints for the operational indices of concentrate grade and the tail grade $y(k) = [y_1(k), y_2(k)]^T$ that yield best possible performance.

The goal in controlling the industrial flotation process (1) is to determine the control input $u(k)$ to make the concentrate grade and the tail grade $y(k)$ track prescribed setpoints $y^*(k)$ determined by Technical Department Personnel in the factory. It is furthermore required that the flotation cell operates safely, specifically the control inputs satisfy maximum allowed physical constraints. For instance, when the pulp height control is too high, the pulp will break out from the flotation cell, which leads to the industrial process being shutdown.

The flotation process operational optimal control problem can be described as the following constrained optimization problem.

$$\begin{aligned} \min_{u(k)} \quad & J(k) = \sum_{l=k+1}^{\infty} \gamma^{l-k-1}(y^* - y(l))^T Q (y^* - y(l)) \\ s.t. \quad & \qquad y(k+1) = F(y(k), u(k)) \\ & \qquad u_{i\min} \leq u_i(k) \leq u_{i\max}, i = 1,2 \end{aligned} \tag{2}$$

where, $0 < \gamma \leq 1$ is a scalar discount factor. The discount factor is a standard accepted means of placing greater emphasize as making small the tracking errors in the immediate future, opposed to the long term errors. Q is a symmetric positive weighting matrix, vector $y^* = [y_1^*, y_2^*]^T \in R^2$ denotes the prescribed setpoints of concentrate grade and tail grade. The flotation process dynamic ore grade function $F(\cdot)$ is an unknown nonlinear function, determined by the flotation industrial process, $u_{i\min}$ and $u_{i\max}$ denote the control input physical constraints.

C. Optimal Tracking Problem Formulation

The constrained optimization problem (2) is difficult to solve in the case of unknown dynamics. It is desired here to develop a method to solve this problem online in real time using data measured along the system trajectories and without knowing the dynamics. To achieve this, problem (2) is reformulated by introducing the dynamical performance index (3).

$$\begin{aligned} V(k) = & \sum_{l=k+1}^{\infty} \gamma^{l-k-1} [(y^* - y(l))^T Q (y^* - y(l)) \\ & + 2\int_{\frac{u_{\min}+u_{\max}}{2}}^{u(l)} \tanh^{-T}(\bar{U}^{-1}(s - \frac{u_{\min}+u_{\max}}{2}))\bar{U}Sds] \\ \equiv & \sum_{l=k+1}^{\infty} \gamma^{l-k-1} [X^T(l) Q_1 X(l) + W(u(l))] \\ \equiv & \sum_{l=k+1}^{\infty} \gamma^{l-k-1} \rho(k) \end{aligned} \tag{3}$$

This performance index captures the performance index $J(k)$ in (2) and also the control constraints there. Specifically, to guarantee the control input $u(k)$ satisfies the physical constraints, the penalty function involving $\tanh^{-T}(\cdot)$ is included in the performance index. It is shown in [33]–[36] that this penalty function guarantees that the controls are bounded according to $u(k) = \bar{U}\tanh(\cdot) + \frac{u_{\min}+u_{\max}}{2}$. This is rigorously developed in next Section. As for the detailed notation in (3), for our 2-input case, $\bar{U} = diag\{\frac{u_{1\max}-u_{1\min}}{2}, \frac{u_{2\max}-u_{2\min}}{2}\}$, S is a positive symmetric matrix. Then it can be shown that $W(u(l))$ is positive when $u(l) \neq \frac{u_{\min}+u_{\max}}{2}$, and $W(u(l))$ is zero when $u(l) = \frac{u_{\min}+u_{\max}}{2}$. Finally, $X(i) = \left[y^T(i), y^{*T}(i)\right]^T$, $Q_1 = C^T Q C$ and $C = \left[\begin{array}{cc} -I & I \end{array}\right]$.

The control objective now is to select the controls $u(k)$ in (1) to minimize the performance index (3).

Remark 1. The performance index (3) includes the minimization objective of problem (2) and also the input constraint. The result is a dynamic optimization that can easily be solved online using reinforcement learning method. Note that in (3) if $y^*(k)$ does not go to zero, the requried control $u(k)$ will not go to zero. Hence, one requires $0 < \gamma < 1$. By choosing a proper discount factor γ and weighting matrix Q in the performance index (3), one can make the tracking error as small as desired.

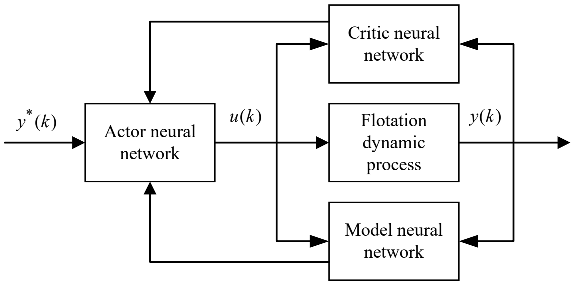


Fig. 4. Flotation industrial process operational optimal control strategy.

III. Real-time Solution of Operational Optimal Control Problem

To solve the optimal control problem of computing controls $u(k)$ in (1) to minimize the performance index (3), standard solution techniques cannot be used since the flotation process dynamic ore grade function $F(\cdot)$ is unknown, and also the presence of the nonlinear function $W(u(l))$ means that the value function $V(k)$ is not quadratic. We shall overcome these difficulties in this section of the paper. In this section, first we provide the ideal offline solution to the optimal control problem of minimizing (3). Then, by using Reinforcement Learning Policy Iteration (PI) techniques we provide a means of solving this problem online using data measured in real time $u(k)$. The result is a model-free algorithm for data-driven optimization (DDO).

A. Operational Optimal Control Strategy

The key to RL is to develop an equation satisfied by the value function $V(k)$, and use to compute the optimal control. This is accomplished in next Section. Then, it is shown how to implement this method in real time using Value Function Approximation (VFA). This method is based on three neural networks (NN). As is standard in RL, one 'critic' NN is used to approximate the value function $V(k)$ in (3). A second 'actor' NN is used to generate the optimal control $u(k)$ in (1). A third model NN is required to identify the nonlinear function $F(\cdot)$ in (1). The overall flotation industrial process operational optimal control strategy is illustrated in Fig. 4.

B. Ideal Optimal Solution

In this section we develop the basic equations for reinforcement learning and provide the ideal offline solution to the operational optimal control solution [20]. A theorem is given to prove the performance of this ideal control solution.

The performance index (3) evaluated at fixed feedback control policies $u(k)$ is known as the Value Function. The first step in developing a RL solution for operational optimal control is to write down an equation that is satisfied by the value function $V(k)$ in (3). A difference equation equivalent to (3) is given by the so-called Bellman equation

$$V(k) = \rho(k) + \gamma V(k+1) \tag{4}$$

This is a consistency equation satisfied by the value function. In terms of this Bellman equation, the Hamilton function of the nonlinear optimal tracking problem (3) is defined as

$$H(k) = V(k) - \rho(k) - \gamma V(k+1) \tag{5}$$

According to Bellman's optimality principle, the optimal value function satisfies the discrete-time Hamilton-Jacobi-Bellman equation

$$V^*(k) = \underset{u(k)}{\arg\min}(\rho(k) + \gamma V^*(k+1)) \tag{6}$$

A necessary condition for optimality is the stationarity condition [20] which is given by

$$\begin{aligned}
&\left.\frac{\partial H(k)}{\partial u(k)}\right|_{u(k)=u^*(k)} = \left.\frac{\partial V(k) - \partial\rho(k) - \gamma\partial V(k+1)}{\partial u(k)}\right|_{u(k)=u^*(k)} \\
&= -2\tanh^{-T}(\bar{U}^{-1}(u(i) - \frac{u_{\min} + u_{\max}}{2}))\bar{U}S\Big|_{u(k)=u^*(k)} \\
&+ \left.\frac{\partial V(k)}{\partial u(k)}\right|_{u(k)=u^*(k)} - \gamma \left.\frac{\partial X^T(k+1)}{\partial u(k)}\frac{\partial V(k+1)}{\partial X(k+1)}\right|_{u(k)=u^*(k)} = 0
\end{aligned} \tag{7}$$

Therefore, the optimal control solution is obtained as

$$\begin{aligned}
u^*(k) =& \bar{U}\tanh(-\frac{\gamma}{2}(\bar{U}S)^{-T}\frac{\partial X^T(k+1)}{\partial u^*(k)}\frac{\partial V(k+1)}{\partial X(k+1)}) \\
&+ \frac{u_{\min} + u_{\max}}{2}
\end{aligned} \tag{8}$$

where $u_{\min} = [u_{1\min}, u_{2\min}]^T$, $u_{\max} = [u_{1\max}, u_{2\max}]^T$.

By substituting the optimal control (8) into the discrete-time Hamilton-Jacobi-Bellman equation (6), the discrete-time optimal Hamilton-Jacobi-Bellman equation becomes

$$V^*(k) = X^T(k)Q_1X(k) + W(u^*(k)) + \gamma V^*(k+1) \tag{9}$$

The ideal solution to the optimal control problem is given by solving (9) for then computing $u^*(k)$ from (8).

The next result shows that the ideal optimal controller we have just derived does indeed solve the optimal tracking problem in (3).

Theorem 1. *Given system* (1) *with the ideal control* (8) *and* (9)*, the ideal optimal control* (8) *minimizes the value function* (3) *and satisfies the constrants in* (2).

Proof: Define that $H^*(k) = H(k)|_{u(k)=u^*(k)}$. For any admissible control input , we can write the value function $V(k)$ in (3) as

$$\begin{aligned}
V(k) &= \sum_{l=k+1}^{\infty}\gamma^{l-k-1}\rho(k) \\
&= \sum_{l=k+1}^{\infty}\gamma^{l-k-1}[V(k) - \gamma V(k+1) + H(k)] \\
&= \sum_{l=k+1}^{\infty}\gamma^{l-k-1}H(k) + V(k)
\end{aligned} \tag{10}$$

Note that,

$$\begin{aligned}
H^*(k) - H(k) =& V^*(k) - \rho^*(k) - \gamma V^*(k+1) \\
&- V(k) + \rho(k) + \gamma V(k+1) \\
=& -\gamma V^*(k+1) + \gamma V(k+1) \\
&+ 2\int_{u^*(k)}^{u(k)}\tanh^{-T}(\bar{U}^{-1}(s - \frac{u_{\min} + u_{\max}}{2}))\bar{U}Sds
\end{aligned} \tag{11}$$

Since $H^*(k) = 0$, (11) can be written as

$$\begin{aligned}
V(k) = V^*(k) + \sum_{l=k+1}^{\infty}\gamma^{l-k-1}\Big[&\gamma V^*(k+1) - \gamma V(k+1) \\
&- 2\int_{u^*(k)}^{u(k)}\tanh^{-T}(\bar{U}^{-1}(s - \frac{u_{\min} + u_{\max}}{2}))\bar{U}Sds\Big]
\end{aligned} \tag{12}$$

Taking $\frac{u_{\min}+u_{\max}}{2}$ and $\bar{U}$ to the left side of (8), then taking $\tanh^{-1}(\cdot)$ from two sides of (8), yield

$$2(\bar{U}S)^T\tanh^{-1}(\bar{U}^{-1}(u^*(k) - \frac{u_{\min} + u_{\max}}{2})) = -\gamma\frac{\partial V(k+1)}{\partial u(k)} \tag{13}$$

By integrating (13) one has

$$\begin{aligned}
&2\int_{u^*(k)}^{u(k)}(\bar{U}S)^T\tanh^{-1}(\bar{U}^{-1}(u^*(k) - \frac{u_{\min} + u_{\max}}{2}))ds \\
&= -\gamma\int_{u^*(k)}^{u(k)}\frac{\partial V(k+1)}{\partial u(k)} = \gamma V^*(k+1) - \gamma V(k+1)
\end{aligned} \tag{14}$$

The equation above is equivalently to

$$\begin{aligned}
&2(u(k) - u^*(k))(\bar{U}S)^T\tanh^{-1}(\bar{U}^{-1}(u^*(k) - \frac{u_{\min} + u_{\max}}{2})) \\
&= \gamma V^*(k+1) - \gamma V(k+1)
\end{aligned} \tag{15}$$

By substituting (15) into (12)

$$\begin{aligned}
V(k) = V^*(k) + \sum_{l=k+1}^{\infty}\gamma^{l-k-1}\Big[&2(u(k) - u^*(k))(\bar{U}S)^T \\
&\tanh^{-1}(\bar{U}^{-1}(u^*(k) - \frac{u_{\min} + u_{\max}}{2})) \\
&-2\int_{u^*(k)}^{u(k)}\tanh^{-T}(\bar{U}^{-1}(s - \frac{u_{\min} + u_{\max}}{2}))\bar{U}Sds\Big] \\
\equiv& Y(u(k)) + V^*(k)
\end{aligned} \tag{16}$$

Using paper [23], we can prove that $Y(u(k)) > 0$ when $u(k) \neq u^*(k)$, $Y(u(k)) = 0$ when $u(k) = u^*(k)$, which is equivalently to $V(k) > V^*(k)$ when $u(k) \neq u^*(k)$, $V(k) = V^*(k)$ when $u(k) = u^*(k)$.

In (8), the range of $\tanh(\cdot)$ is -1 to 1, so the range of the optimal control $u^*(k)$ is $-\bar{U} + \frac{u_{\min}+u_{\max}}{2}$ to $\bar{U} + \frac{u_{\min}+u_{\max}}{2}$, thus the optimal solution satisfies the physical constrains in (2).

The proof is complete. □

C. Real-time Implementation of Operational Optimal Control Using Policy Iteration

In this section we develop a reinforcement learning controller for computing online the optimal control $u(k)$ that minimizes (3) online using data measured in real time along the system trajectories and without knowing the system dynamics.

This is accomplished by a technique that employs Value Function Approximation using a critic neural network, control generation using a second actor NN, and system identification using a third NN.

1) Policy Iteration and the Critic Neural Network: The ideal solution to the optimal control problem given by (9) and (8) cannot be found because the solution $V^*(k)$ to the HJB equation is generally impossible to find analytically due to the nonlinearity of the dynamics (1) and the performance index (3). A Policy Iteration approach overcomes part of this problem. Algorithm 1 finds the optimal control by repeated solutions of the Bellman equation (4).

Algorithm 1 Policy Iteration Solution for Operational Optimal Control

Initiation: Select a stabilizing initial control $u_0(k-1)$ that satisfies the constraints. Iterate the following two steps on i until convergence.

1) **Policy Evaluation**: Find the value function related to the policy $u_i(k-1)$ by solving the Bellman equation (4), that is

$$V_i(k-1) = \rho_i(k-1) + \gamma V_i(k) \tag{17}$$

2) **Policy Improvement**: Update the policy using (8), that is

$$\begin{aligned} u^*_{i+1}(k-1) =& \bar{U}\tanh\left(-\frac{\gamma}{2}(\bar{U}S)^{-T}\frac{\partial X^T(k)}{\partial u^*_i(k-1)}\frac{\partial V_i(k)}{\partial X(k)}\right) \\ &+\frac{u_{\min}+u_{\max}}{2} \end{aligned} \tag{18}$$

To implement this algorithm in real-time a critic neural network is used for Value Function Approximation (VFA) to allow approximate solution of the Bellman equation (17). At each iteration i, the critic NN is

$$L_c\hat{V}_i(k) = W^T_{ci}(k)\sigma(V^T_c\hat{X}(k)) \tag{19}$$

where $\sigma(\cdot)$ is the NN activation function, $\hat{X}(k) = \left[\hat{y}^T(k), y^{*T}(k)\right]^T$, $\sigma(z) = \frac{e^z-e^{-z}}{e^z+e^{-z}}$, V^T_c are the weights of the input layer to the hidden layer and $W^T_{ci}(k)$ are the weights of the hidden layer to the output layer. Constant $L_c > 0$ is used for scaling the value function down so that it can be approximated by NN weights that are not too large. The first-layer weight V^T_c is a constant matrix that can be selected to avoid over-activation in the neurons, or selected randomly. Then, $\sigma(V^T_c\hat{X}(k))$ provides a basic [37].

Note that the critic NN is implemented at sampling time k, thus the critic NN of $\hat{V}_i(k-1)$ is

$$L_c\hat{V}_i(k-1) = W^T_{ci}(k)\sigma(V^T_cX(k-1)) \tag{20}$$

In order to improve the accuracy of the critic NN, we use the input and the output date of the flotation process to train the weights $W^T_{ci}(k)$. Accordingly, define the equation approximation error

$$e_{ci}(k) = L_c\rho_i(k-1) + L_c\gamma\hat{V}_i(k) - L_c\hat{V}_i(k-1) \tag{21}$$

and the squared error

$$E_{ci}(k) = \frac{1}{2}e^2_{ci}(k) \tag{22}$$

The train algorithm is gradient descent, which is

$$W_{c(i+1)}(k) = W_{ci}(k) - l_{ci}(k)\frac{\partial E_{ci}(k)}{\partial W_{ci}(k)} \tag{23}$$

where $l_{ci}(k)$ is the variable learning rate given by

$$l_{c(i+1)}(k) = \exp(-\alpha_c)l_{ci}(k) \tag{24}$$

with α_c is a positive constant. Using the chain rule one obtains

$$\begin{aligned} \frac{\partial E_{ci}(k)}{\partial W_{ci}(k)} =& \frac{\partial E_{ci}(k)}{\partial e_{ci}(k)}\frac{\partial e_{ci}(k)}{\partial \hat{V}_i(k)}\frac{\partial \hat{V}_i(k)}{\partial W_{ci}(k)} \\ &+\frac{\partial E_{ci}(k)}{\partial e_{ci}(k)}\frac{\partial e_{ci}(k)}{\partial \hat{V}_i(k-1)}\frac{\partial \hat{V}_i(k-1)}{\partial W_{ci}(k)} \\ =& \gamma\sigma(V^T_c\hat{X}(k))e^T_{ci}(k) - \sigma(V^T_cX(k-1))e^T_{ci}(k) \end{aligned} \tag{25}$$

Thus, the critic NN training algorithm becomes

$$\begin{aligned} &W_{c(i+1)}(k) = W_{ci}(k) \\ &-l_{ci}(k)\left[\gamma\sigma(V^T_c\hat{X}(k))e^T_{ci}(k) - \sigma(V^T_cX(k-1))e^T_{ci}(k)\right] \end{aligned} \tag{26}$$

2) Actor Neural Network: The control update at each step in Algorithm 1 is (18). To compute the optimal control action at time k, we employ a second actor NN given by

$$L_a\hat{u}_i(k) = W^T_{ai}(k)\sigma(V^T_aX(k)) \tag{27}$$

where V^T_a are the weights of the input layer to the hidden layer and $W^T_{ai}(k)$ are the weights of the hidden layer to the output layer. Constant $L_a > 0$ is used for scaling the value function down so that it can be approximated by NN weights that are not too large. Similarly, the first-layer weight V^T_a is a constant matrix that can be selected to avoid over-activation in the neurons, or selected randomly. Then, $\sigma(V^T_aX(k))$ provides a basic [37].

Then the squared error is defined as

$$E_{ai}(k) = \frac{1}{2}e^T_{ai}(k)e_{ai}(k) \tag{28}$$

In (28), the error is

$$e_{ai}(k) = L_a\left[\hat{u}_i(k-1) - u^*_i(k-1)\right] \tag{29}$$

where $u^*_i(k-1)$ is defined in (18), calculate gradients as

$$\begin{aligned} \frac{\partial X^T(k)}{\partial u^*_i(k-1)} =& \frac{\partial X^T(k)}{\partial y(k)}\frac{\partial F(y(k-1),u^*_i(k-1))}{\partial u^*_i(k-1)} \\ =& I_1\frac{\partial F(y(k-1),u^*_i(k-1))}{\partial u^*_i(k-1)} \end{aligned} \tag{30}$$

$$\frac{\partial \hat{V}_i(k)}{\partial \hat{X}(k)} = L^{-1}_cW^T_{ci}(k)V^T_c\dot{\sigma}(V^T_c\hat{X}(k)) \tag{31}$$

where $I_1 = diag\{I_{2\times2}, 0_{2\times2}\}$, $I_2 = [0_{2\times2}, I_{2\times2}]$.
The train algorithm is gradient descent, which is

$$W_{a(i+1)}(k) = W_{ai}(k) - l_{ai}(k)\frac{\partial E_{ai}(k)}{\partial W_{ai}(k)} \tag{32}$$

where $l_{ai}(k)$ is the variable learning rate given by

$$l_{a(i+1)}(k) = \exp(-\alpha_a) l_{ai}(k) \tag{33}$$

with α_a is a positive constant. Using the chain rule one obtains

$$\begin{aligned} \frac{\partial E_{ai}(k)}{\partial W_{ai}(k)} &= \frac{\partial E_{ai}(k)}{\partial e_{ai}(k)} \frac{\partial e_{ai}(k)}{\partial \hat{u}_i(k-1)} \frac{\partial \hat{u}_i(k-1)}{\partial W_{ai}(k)} \\ &= \sigma(V_a^T X(k-1)) e_{ai}^T(k) \end{aligned} \tag{34}$$

Thus, the actor NN training algorithm becomes

$$W_{a(i+1)}(k) = W_{ai}(k) - l_{ai}(k)\sigma(V_a^T X(k-1)) e_{ai}^T(k) \tag{35}$$

Remark 2. Finding an initial stabilizing control is a common problem in PI. In a real factory, the human operator (Technical Department Personnel) can determine the pulp height and feed flow using his previous experience and process knowledge to satisfy the constraints. Thus, an initial stabilizing but nonoptimal $u_0(k)$ can be obtained for real flotation processes by operator experience.

IV. A New Interleaved Learning Algorithm for Online Solution of the OOC Problem

In this section we develop a reinforcement learning technique known as Interleaved Learning (IL) that combines the best features of Policy Iteration and Value Iteration for real-time operational optimal control that performs better than policy iteration given in Algorithm 1.

A. Interleaved Learning

In standard Policy Iteration, one trains the critic NN repeatedly using (26) until (17) is satisfied to some accuracy. The training termination condition is $\left|\hat{V}_{i+1}(k-1) - \hat{V}_i(k-1)\right| \le \varepsilon_c$, with ε_c is a small positive constant. Then the action NN is trained repeatedly using (35) until (18) is satisfied. This results in slow convergence at each step and possibly long transients in the flotation process. An alternative is Value Iteration where (17) is replaced by the one-step update

$$\hat{V}_{i+1}(k-1) = \rho_i(k-1) + \gamma \hat{V}_i(k) \tag{36}$$

which uses the old NN weighs on the right-hand side. Then the action NN is trained repeatedly using (35) until (18) is satisfied. It is generally known that PI converges faster that VI [38], yet it requires repeated solutions of the Bellman equation (17), which is a nonlinear Lyapunov equation. The control is only updated after this equation is satisfied to some accuracy. This can result in slow control updates. On the other hand, VI updates the control more often, but convergence to the optimal solution is slow. For these reasons a whole family of generalized policy iteration algorithms have been proposed that combine aspects of PI and VI [38]. Yet, proofs for the convergence of PI are generally not available. In this paper, the proposed algorithm is a kind of generalized Policy Iteration that interleaves single-step updates of the critic value function and the control policy. It is seen in simulations that this algorithm has better performance than either PI or VI, and moreover, we are able to provide a rigorous proof of convergence.

To correct these deficiencies in PI and VI and provide a practically useful Interleaved Learning procedure for implementing optimal learning online for industrial processes, we propose a new algorithm which makes only one step in the policy evaluation (17) and then makes one step in the policy improvement (18). That is, the training algorithms (26) and (35) are interleaved. This procedure is a variant of Generalized Policy Iteration [38], and is shown in Fig. 5.

The next main result verifies the convergence of Interleaved Learning.

Theorem 2. *Given neural network* (19) *and* (27)*, applying the training algorithm* (23) *and* (32) *once at each iteration in i. Select $l_{c0}(k)$ and $l_{a0}(k)$ to satisfy*

$$0 < l_{a0}(k) < \frac{1}{4a_a(k)} \tag{37}$$

$$0 < l_{c0}(k) < \frac{1}{4a_c(k)} \tag{38}$$

where $a_a(k)$ and $a_c(k)$ are defined in the proof. Then the neural networks (19) *and* (27) *converge to the neighborhood of the solutions to* (17) *and* (18).

Proof: Since V_c and V_a are randomly selected, $\sigma(V_c^T X(k))$ and $\sigma(V_a^T X(k))$ are the basic [37], the neural network estimation error can be expressed by

$$L_c V_i(k-1) = W_{ci}^{*T}(k)\sigma(V_c^T X(k-1)) + \varepsilon_{ci}(X(k-1)) \tag{39}$$

$$L_c V(k-1) = W_c^{*T}(k)\sigma(V_c^T X(k-1)) + \varepsilon_c(X(k-1)) \tag{40}$$

$$L_a u_i^*(k-1) = W_{ai}^{*T}(k)\sigma(V_a^T X(k-1)) + \varepsilon_{ai}(X(k-1)) \tag{41}$$

$$L_a u^*(k-1) = W_a^{*T}(k)\sigma(V_a^T X(k-1)) + \varepsilon_a(X(k-1)) \tag{42}$$

where $W_{ci}^{*T}(k)$, $W_{ai}^{*T}(k)$, $W_c^{*T}(k)$ and $W_a^{*T}(k)$ are the ideal weight parameters, $\varepsilon_{ci}(X(k-1))$, $\varepsilon_{ci}(X(k-1))$, $\varepsilon_c(X(k-1))$ and $\varepsilon_a(X(k-1))$ are the reconstruction error and bounded in the compact sets, $|\varepsilon_{ci}(X(k-1))| \le \varepsilon_{ciM}$, $\|\varepsilon_{ai}(X(k-1))\|_2 \le \varepsilon_{aiM}$, $|\varepsilon_c(X(k-1))| \le \varepsilon_{cM}$ and $\|\varepsilon_a(X(k-1))\|_2 \le \varepsilon_{aM}$.

Refer to [39], one obtains $\lim_{i\to\infty} W_{ci}^{*T}(k) = W_c^{*T}(k)$ and $\lim_{i\to\infty} W_{ai}^{*T}(k) = W_a^{*T}(k)$, so there must exist $W_c > 0$ and $W_a > 0$, such that $\|W_{ci}^{*T}(k) - W_c^{*T}(k)\|_2 \le W_c$ and $\|W_{ai}^{*T}(k) - W_a^{*T}(k)\|_2 \le W_a$ hold, then define $\bar{W}_{ci}^{*T}(k) = W_{ci}^{*T}(k) - W_c^{*T}(k)$ and $\bar{W}_{ai}^{*T}(k) = W_{ai}^{*T}(k) - W_a^{*T}(k)$.

Consider the following Lyapunov function,

$$\begin{aligned} L(\hat{W}_{ci}(k), \hat{W}_{ai}(k)) &= L_1(\hat{W}_{ci}(k)) + L_2(\hat{W}_{ai}(k)) \\ &= tr\{\hat{W}_{ci}^T(k)\hat{W}_{ci}(k)\} + tr\{\hat{W}_{ai}^T(k)\hat{W}_{ai}(k)\} \end{aligned} \tag{43}$$

where $\hat{W}_{ci}^T(k) = W_{ci}^T(k) - W_c^{*T}(k)$ and $\hat{W}_{ai}^T(k) = W_{ai}^T(k) - W_a^{*T}(k)$.

Then, the difference of the Lyapunov function is

$$\begin{aligned} \Delta L_1 &= tr\{\hat{W}_{c(i+1)}^T(k)\hat{W}_{c(i+1)}(k)\} - tr\{\hat{W}_{ci}^T(k)\hat{W}_{ci}(k)\} \\ &= -2l_{ci}(k)\hat{W}_{ci}^T(k)\sigma_1(k)e_{ci}^T(k) \\ &\quad + l_{ci}^2(k)e_{ci}(k)\sigma_1^T(k)\sigma_1(k)e_{ci}^T(k) \end{aligned} \tag{44}$$

$$\begin{aligned}\Delta L_2 &= tr\{\hat{W}^T_{a(i+1)}(k)\hat{W}_{a(i+1)}(k)\} - tr\{\hat{W}^T_{ai}(k)\hat{W}_{ai}(k)\} \\ &= -2l_{ai}(k)\hat{W}^T_{ai}\sigma_2(k)e^T_{ai}(k) \\ &\quad + l^2_{ai}(k)e_{ai}(k)\sigma^T_2(k)\sigma_2(k)e^T_{ai}(k)\end{aligned} \tag{45}$$

where $\sigma_1(k) = \gamma\sigma(V^T_c\hat{X}(k)) - \sigma(V^T_c X(k-1))$, $\sigma_2(k) = \sigma(V^T_a X(k-1))$.

Refer to paper [40], one obtains

$$\begin{aligned}e_{ci}(k) &= (\hat{W}^T_{ci}(k) - \bar{W}^{*T}_{ci}(k))\sigma_1(k) - \gamma\varepsilon_{ci}(X(k)) + \varepsilon_{ci}(X(k-1)) \\ &= (\hat{W}^T_{ci}(k) - \bar{W}^{*T}_{ci}(k))\sigma_1(k) - \varepsilon_{ciH}(k)\end{aligned} \tag{46}$$

$$e_{ai}(k) = (\hat{W}^T_{ai}(k) - \bar{W}^{*T}_{ai}(k))\sigma_2(k) - \varepsilon_{ai}(X(k-1)) \tag{47}$$

where $|\varepsilon_{ciH}(k)| \leq \varepsilon_{cHM} = (1+\gamma)\varepsilon_{cM}$.

Inspired by [41], [42], using Frobenius norm and applying CS inequality $(a_1 + \cdots + a_n)^2 \leq na^2_1 + \cdots + na^2_n$. And define

$$a_c(k) = \sigma^T_1(k)\sigma_1(k)$$

$$a_a(k) = \sigma^T_2(k)\sigma_2(k)$$

the difference of the Lyapunov function becomes

$$\begin{aligned}\Delta L_1 \leq & -2l_{ci}(k)\|\hat{W}^T_{ci}(k)\|^2_2 a_c(k) + 2l_{ci}(k)\hat{W}^T_{ci}(k)\sigma_1(k)\varepsilon_{ciH} \\ & + 3l^2_{ci}(k)\|\hat{W}^T_{ci}(k)\|^2_2 a^2_c(k) + 3l^2_{ci}(k)a_c(k)\varepsilon^2_{ciH} \\ & + 3l^2_{ci}(k)W^2_c(k)a^2_c(k) + 2l_{ci}(k)\hat{W}^T_{ci}(k)\bar{W}_{ci}(k)a_c(k) \\ \leq & -(l_{ci}(k)a_c(k) - 4l^2_{ci}(k)a^2_c(k))\|\hat{W}^T_{ci}(k)\|^2_2 + l_{ci}(k)\varepsilon^2_{cHM} \\ & + 3l^2_{ci}(k)a_c(k)\varepsilon^2_{cHM} + W^2_c + 3l^2_{ci}(k)a^2_c(k)W^2_c\end{aligned} \tag{48}$$

$$\begin{aligned}\Delta L_2 \leq & -2l_{ai}(k)\|\hat{W}^T_{ai}(k)\|^2_2 a_a(k) + 3l^2_{ai}(k)\|\hat{W}^T_{ai}(k)\|^2_2 a^2_a(k) \\ & + 2l_{ai}(k)\hat{W}^T_{ai}(k)\sigma_2(k)\varepsilon_{ai}(k-1) + 3l^2_{ai}(k)W^2_a(k)a^2_a(k) \\ & + 3l^2_{ai}(k)a_a(k)\|\varepsilon_{ai}(k-1)\|^2_2 + 2l_{ai}(k)\hat{W}^T_{ai}(k)\bar{W}_{ai}(k)a_a(k) \\ \leq & -(l_{ai}(k)a_a(k) - 4l^2_{ai}(k)a^2_a(k))\|\hat{W}^T_{ai}(k)\|^2_2 + l_{ai}(k)\varepsilon^2_{aM} \\ & + 3l^2_{ai}(k)a_a(k)\varepsilon^2_{aM} + W^2_a + 3l^2_{ai}(k)a^2_a(k)W^2_a\end{aligned} \tag{49}$$

To obtain that the difference of the Lyapunov function is negtive, one obtains

$$\Delta^1_{i,k} = l_{ci}(k)a_c(k) - 4l^2_{ci}(k)a^2_c(k) > 0 \tag{50}$$

$$\Delta^2_{i,k} = l_{ai}(k)a_a(k) - 4l^2_{ai}(k)a^2_a(k) > 0 \tag{51}$$

Then we can select $l_{c0}(k)$ and $l_{a0}(k)$ to satisfy

$$0 < l_{a0}(k) < \frac{1}{4a_a(k)} \tag{52}$$

$$0 < l_{c0}(k) < \frac{1}{4a_c(k)} \tag{53}$$

The first difference $\Delta L_1 \leq 0$ as long as $\|\hat{W}^T_{ci}(k)\|_2 > \sqrt{(l_{ci}(k) + 3l^2_{ci}(k)a_c(k))\varepsilon^2_{cHM} + W^2_c + 3l^2_{ci}(k)a^2_c(k)W^2_c} \div \sqrt{\Delta^1_{i,k}} = B_c$, and the second difference $\Delta L_2 \leq 0$ as long as $\|\hat{W}^T_{ai}(k)\|_2 > \sqrt{(l_{ai}(k) + 3l^2_{ai}(k)a_a(k))\varepsilon^2_{aM} + W^2_a + 3l^2_{ai}(k)a^2_a(k)W^2_a} \div \sqrt{\Delta^2_{i,k}} = B_a$. By Lyapunov theorem, the critic NN weight estimation error $\|\hat{W}^T_{ci}(k)\|_2$ and actor NN weight estimation error $\|\hat{W}^T_{ai}(k)\|_2$ are ultimately bounded.

The convergence of the estimated value function and control input to their respective optimal values is shown as

$$|\hat{V}_i(k-1) - V(k-1)| \leq L^{-1}_c(B_c\|\sigma(V^T_c X(k-1))\|_2 + \varepsilon_{cM}) \tag{54}$$

$$\|L_a(\hat{u}_i(k-1) - u^*(k-1))\|_2 \leq (B_a\|\sigma_2(k)\|_2 + \varepsilon_{aM}) \tag{55}$$

The proof is completed. □

Remark 3. The convergence neighborhood in this proof depends on two parts, namely $W_c = \left\|W^{*T}_{ci}(k) - W^{*T}_c(k)\right\|_2$ and $\varepsilon_{ci}(X(k-1))$. For the first part, when the iteration step i goes to infinity, then $\lim_{i\to\infty} W^{*T}_{ci}(k) - W^{*T}_c(k) = 0$; for the second part, the bound of $\varepsilon_{ci}(X(k-1))$ depends on the number of NN hidden layer nodes, when the number of the NN hidden layer nodes becomes larger, the bound of $\varepsilon_{ci}(X(k-1))$ goes to zero. Thus, we can shrink the region of the convergence neighborhood by selecting the number of NN hidden layer nodes and setting the iteration step i.

B. System Identification Neural Network

In (18), $\frac{\partial X^T(k)}{\partial u(k-1)}$ depends on the dynamic model function $F(\cdot)$ in (1), but $F(\cdot)$ is unknown, so the optimal control solution cannot be obtained by using (18) directly.

The model of the single-cell flotation process is explained in Section II. A. The output $y(k)$ and the input $u(k)$ are both 2-vectors. The output $y(k)$, the input $u(k)$ can be measured. Therefore, introduce a model NN (see Fig. 4) to approximate the flotation process dynamic ore grade function $F(\cdot)$

$$L_m\hat{y}_i(k) = W^T_{mi}(k)\sigma(V^T_m\Phi(k-1)) \tag{56}$$

where $\Phi(k-1) = [y^T(k-1), u^T(k-1)]^T \in R^4$ is the input of the model neural network, V^T_m are the weights of the input layer to the hidden layer and $W^T_{mi}(k)$ are the weights of the hidden layer to the output layer. Constant $L_m > 0$ is used for scaling the value function down so that it can be approximated by NN weights that are not too large. Similarly, the first-layer weight V^T_m is a constant matrix that can be selected to avoid over-activation in the neurons, or selected randomly. Then, $\sigma(V^T_m\Phi(k))$ provides a basic [37].

In order to improve the accuracy of the model NN, we use the input and the output data of the flotation process to train the model neural network, the training algorithm is gradient descent

$$W_{m(i+1)}(k) = W_{mi}(k) - l_{mi}(k)\frac{\partial E_{mi}(k)}{\partial W_{mi}(k)} \tag{57}$$

The $E_{mi}(k)$ is defined as

$$\begin{aligned}E_{mi}(k) &= \frac{1}{2}e^T_{mi}(k)e_{mi}(k) \\ e_{mi}(k) &= L_m[\hat{y}_i(k) - y(k)]\end{aligned} \tag{58}$$

where $l_{mi}(k)$ is the variable learning rate given by

$$l_{m(i+1)}(k) = \exp(-\alpha_m)l_{mi}(k) \tag{59}$$

with α_m is a positive constant. Using the chain rule one obtains

$$\begin{aligned}\frac{\partial E_{mi}(k)}{\partial W_{mi}(k)} &= \frac{\partial E_{mi}(k)}{\partial e_{mi}(k)}\frac{\partial e_{mi}(k)}{\partial \hat{y}_i(k)}\frac{\partial \hat{y}_i(k)}{\partial W_{mi}(k)} \\ &= \sigma(V_m^T\Phi(k-1))e_{mi}^T(k)\end{aligned} \tag{60}$$

Thus, the model NN training algorithm becomes

$$W_{m(i+1)}(k) = W_{mi}(k) - l_{mi}(k)\sigma(V_m^T\Phi(k-1))e_{mi}^T(k) \tag{61}$$

The termination condition is

$$\|e_{mi}(k)\|_2 \leq \varepsilon_m \tag{62}$$

Theorem 3. *Given neural network* (56)*, applying the training algorithm* (57)*, select* $l_{m0}(k)$*to satisfy*

$$0 < l_{m0}(k) < \frac{1}{2a_m(k)} \tag{63}$$

where $a_m(k)$ *is defined in the proof. Then the neural network weights in* (56) *are convergent.*

Proof: Since V_m is randomly selected, $\sigma(V_m^T\Phi(k-1))$ is the basic [37], the neural network estimation error can be expressed by

$$L_m y(k) = W_m^{*T}(k)\sigma(V_m^T\Phi(k-1)) + \varepsilon_m(\Phi(k-1)) \tag{64}$$

where W_m^{*T} are the ideal weight parameters, $\varepsilon_m(\Phi(k-1))$ is the reconstruction error and bounded in the compact set, $\|\varepsilon_m(\Phi(k-1))\|_2 \leq \varepsilon_{mM}$.

Consider the following Lyapunov function,

$$L_m(\hat{W}_{mi}(k), \hat{W}_{mi}(k)) = tr\{\hat{W}_{mi}^T(k)\hat{W}_{mi}(k)\} \tag{65}$$

Then, the difference of the Lyapunov function is

$$\begin{aligned}\Delta L_m &= tr\{\hat{W}_{m(i+1)}^T(k)\hat{W}_{m(i+1)}(k)\} - tr\{\hat{W}_{mi}^T(k)\hat{W}_{mi}(k)\} \\ &= -2l_{mi}(k)\hat{W}_{mi}^T\sigma(V_m^T\Phi(k-1))e_{mi}^T(k) \\ &\quad + l_{mi}^2(k)e_{mi}(k)\sigma^T(V_m^T\Phi(k-1))\sigma(V_m^T\Phi(k-1))e_{mi}^T(k)\end{aligned} \tag{66}$$

Define

$$a_m(k) = \sigma^T(V_m^T\Phi(k-1))\sigma(V_m^T\Phi(k-1))$$

the difference of the Lyapunov function becomes

$$\begin{aligned}\Delta L_m \leq &-(l_{mi}(k)a_m(k) - 2l_{mi}^2(k)a_m^2(k))\|\hat{W}_{mi}^T(k)\|_2^2 \\ &+ l_{mi}(k)\varepsilon_{mM}^2 + 2l_{mi}^2(k)a_m(k)\varepsilon_{mM}^2\end{aligned} \tag{67}$$

To obtain that the difference of the Lyapunov function is negtive

$$l_{mi}(k)a_m(k) - 2l_{mi}^2(k)a_m^2(k) > 0 \tag{68}$$

Then we can select $l_{m0}(k)$ to satisfy

$$0 < l_{m0}(k) < \frac{1}{2a_m(k)} \tag{69}$$

The difference $\Delta L_m \leq 0$ as long as $\|\hat{W}_{mi}^T\|_2 > \sqrt{(l_{mi}(k)\varepsilon_{mM}^2 + 2l_{mi}^2(k)a_m(k)\varepsilon_{mM}^2)/(l_{mi}(k)a_m(k) - 2l_{mi}^2(k)a_m^2(k))}$ $= B_m$. By Lyapunov theorem, the model NN weight estimation error $\|\hat{W}_{mi}^T\|_2$ is ultimately bounded.

The convergence of the estimated model to its optimal value is shown as

$$\|L_m(\hat{y}_i(k) - y(k))\|_2 \leq (B_m\|\sigma(V_m^T\Phi(k-1))\|_2 + \varepsilon_{mM}) \tag{70}$$

The proof is completed. □

C. Interleaved Learning Algorithm for Real-time Generation of OOC

Based on the previous developments, we deliver the following algorithm for online computation of the operational optimal control solution. The algorithm is a combination of off-line training for system identification and on-line learning for the optimal value function and control policy. The on-line portion interleaves single updates of the critic NN and the actor NN and as such is a form of generalized PI [38]. It is a data-driven optimization algorithm that determines operational optimal controls in real time by using data measured as the process operates.

```
Algorithm 2 Real-time Learning of OOC
Off-line Initial Training.
  Step 1: Using historical input-output process data, train the
          weights of model NN (56) using (61) to initialize
          the model NN. Set $k = 2$;
Online Computations.
Update Identification Model
  Step 2: Set $i = 0$;
  Step 3: Set $i = i + 1$;
  Step 4: Use the measured output $y(k)$, and previous output
          $y(k-1)$ and input $u(k-1)$ to update the weights
          of model NN (56) using (61);
  Step 5: Check $\|e_{mi}(k)\|_2 \leq \varepsilon_m$, if it is not satisfied, go to
          Step 2;
Update Critic and Actor NN
  Step 6: Set $i = 0$, and initialize the weights of critic NN
          (19) and the weights of actor NN (56);
  Step 7: Set $i = i + 1$;
  Step 8: Use the measured output $y(k)$, setpoint $y^*(k)$, and
          previous output $y(k-1)$ and setpoint $y^*(k-1)$ to
          update the weights of critic NN (19) using (26);
  Step 9: Use previous output $y(k-1)$ and setpoint $y^*(k-1)$
          to update the weights of actor NN (27) using (35);
  Step 10: Check $|\hat{V}_i(k-1) - \hat{V}_{i+1}(k-1)| \leq \varepsilon_c$, if it is not
           satisfied, go to Step 7, else get the input $\hat{u}_i(k)$
           using (27) and introduce this input into the flotation
           process;
  Step 11: Set $k = k + 1$, and return to step 2.
```

The algorithm procedure is shown as a flow chart in Fig. 5.

V. Flotation Industrial Process Operational Optimal Control Simulation Experiment

In this section, the model-free data-driven operational optimal control method of Algorithm 2 is applied to a single-cell flotation industrial process in Subsection B. Computer simulation experiments are used to verify the effectiveness of

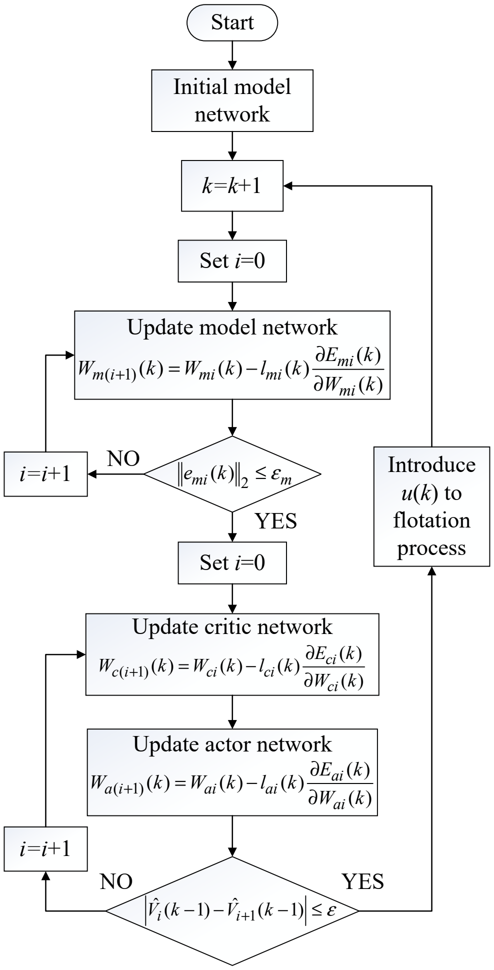

Start
Initial model network
k=k+1
Set i=0
Update model network
$W_{m(i+1)}(k)=W_{mi}(k)-l_{mi}(k)\frac{\partial E_{mi}(k)}{\partial W_{mi}(k)}$
i=i+1
NO
$\left\|e_{mi}(k)\right\|_2 \le \varepsilon_m$
YES
Set i=0
Update critic network
$W_{c(i+1)}(k)=W_{ci}(k)-l_{ci}(k)\frac{\partial E_{ci}(k)}{\partial W_{ci}(k)}$
Update actor network
$W_{a(i+1)}(k)=W_{ai}(k)-l_{ai}(k)\frac{\partial E_{ai}(k)}{\partial W_{ai}(k)}$
i=i+1
NO
YES
$\left|\hat{V}_i(k-1)-\hat{V}_{i+1}(k-1)\right| \le \varepsilon$
Introduce u(k) to flotation process

Fig. 5. Algorithm flow chart.

the proposed method. The results show that this algorithm learns the optimal control solution online and makes the operational indices track the desired values without violating the constraints. Moreover, standard PI and VI are simulated in Subsection C and D, and seen to yield inferior performance.

A. Flotation Process Model Introduction Details

The following assumptions are made on the flotation cell process.

- Constant air flow;
- Two mineralogical classes: rich mineral (class *i*: 1, mainly chalcopyrite or copper ore) and poor mineral (class *i*: 2, mainly gangue);
- Each phase in the cell is perfectly mixed;
- Material transfer takes place between phases in both directions, depending on the flotation rate in the pulp and the drainage rate in the froth.

The dynamic model of single cell flotation process is [43]–[48]

$$\frac{dM_p^i}{dt} = -(k_p^i + \frac{q_T}{Ah_p})M_p^i + k_e^i M_e^i + q_a X_a^i \tag{71}$$

$$\frac{dM_e^i}{dt} = -(k_e^i + \frac{q_c}{A(H-h_p)})M_e^i + k_p^i M_p^i \tag{72}$$

TABLE I
THE PARAMETERS OF THE FLOTATION INDUSTRIAL PROCESS

Variable	Value	Variable	Value
k_p^1	$65.6min^{-1}$	g_a	0.0234
k_p^2	$316min^{-1}$	A	$53.2m^2$
k_e^1	$17.9min^{-1}$	H	$3.2m$
k_e^2	$0.04min^{-1}$	L_{cu}	42.1%
g_{cp}^1	0.417	g_{cp}^2	0.0034

In above model, $i = 1,2$ represent different mineral, major mineral of mineral 1 is copper ore, major mineral of mineral 2 is gangue. In general, the content of mineral 1 of the mineral mud in flotation cell is known, so the content of mineral 2 is

$$X_a^2 = \frac{g_{cp}^1 - g_a}{g_a - g_{cp}^2} X_a^1 \tag{73}$$

In addition, k_p^i is flotation rate, k_e^i is flotation rate, M_p^i is pulp mass, M_e^i is froth mass, q_a is feed flow, h_p is tail flow, the output of flotation process are concentrate grade L_{cg} and tail grade L_{tg}.

The concentrate grade is computed as

$$L_{cg} = \frac{M_e^1 g_{cp}^1 + M_e^2 g_{cp}^2}{M_e^1 + M_e^2} L_{cu} \tag{74}$$

And the tail grade is

$$L_{tg} = \frac{M_p^1 g_{cp}^1 + M_p^2 g_{cp}^2}{M_p^1 + M_p^2} L_{cu} \tag{75}$$

In this simulation, we select the pulp level h_p and feed flow q_a as control input $u = [h_p, q_a]^T$, the operational indices of concentrate grade L_{cg} and tail grade L_{tg} are the output $y = [L_{cg}, L_{tg}]^T$, the setpoints of concentrate grade L_{cg}^* and tail grade L_{tg}^* are $y^* = [L_{cg}^*, L_{tg}^*]^T$.

B. Flotation Industrial Process Operational Optimal Control Simulation Experiment Using Algorithm 2

The parameters of the Sing-cell flotation industrial process are in the table I.

The control parameters of this simulation experiment are given as follows. The sampling interval is $T_d = 30min$. The height of flotation tank is $H = 3.2m$, the power of the feed pump is finite, so that the physical constraints of the flotation industrial process are: $u_{\min} = [1,3]^T$, $u_{\max} = [3,30]^T$. Depending on the priori knowledge of the flotation process, the weights of the input layer to the hidden layer of three neural networks are $V_m^T = diag\{0.1,1,1,0.1\}$, $V_c^T = diag\{0.1,1,0.1,1\}$ and $V_a^T = diag\{0.1,1,0.1,1\}$, so that the neurons over excitation of NNs can be avoided. The scaling constants of the three neural networks are $L_m = diag\{0.1,1\}$, $L_c = 0.1$ and $L_a = diag\{0.1,1\}$. The initial value of the learning rates of three neural networks are $l_{m0}(k) = 0.01$, $l_{c0}(k) = 0.04$ and $l_{ai}(k) = 0.08$, and $\exp(-\alpha_m) = \exp(-\alpha_a) = \exp(-\alpha_c) = 0.95$. $Q = diag\{1,1\}$, $S = diag\{0.1,0.1\}$, $\gamma = 0.6$, note that the norm of weight matrix Q is bigger then S, then the tracking error will be small. Setpoint is $y^* = [17.34; 0.75]^T$. The initial output-layer weights of the model neural network

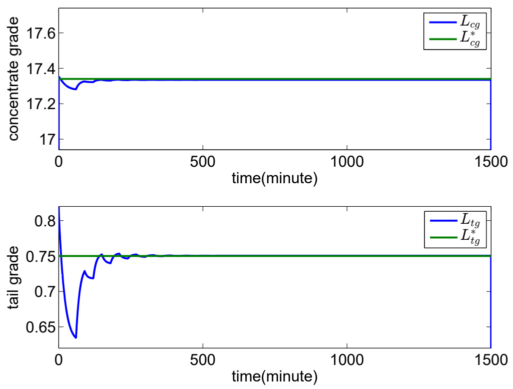


Fig. 6. Flotation process operational index tracking results using Algorithm 2.

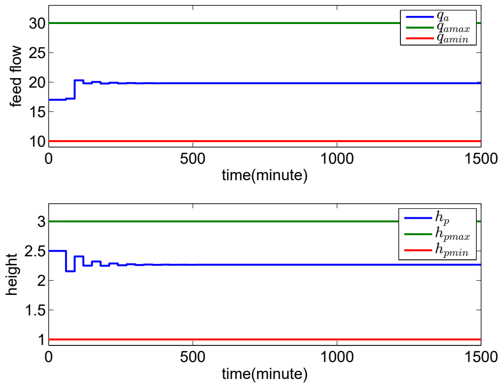


Fig. 7. Flotation process operational optimal control input using Algorithm 2.

$W_{m0}^T(k)$, the critic neural network $W_{c0}^T(k)$, and the actor neural network $W_{a0}^T(k)$ are initialized randomly.

Given the dynamic model and the control parameters, the operational optimal control method of Algorithm 2 is applied to single-cell flotation industrial process. An initial stabilizing control input of $u(1) = [2.8, 17]^T$ is applied for the first two time steps. At 60 min, that is after two time steps, the control generated by Algorithm 2 is switched in. Fig.6 shows the operational index tracking results, the operational indices of flotation industrial process are seen to track the setpoint operational indices. Fig.7 shows the flotation industrial process operational optimal control input, which are seen not to exceed the physical constraints.

Fig. 8, Fig. 9 and Fig. 10 show the output-layer weights of the three neural networks during the control run, which converge to the constant values

$$W_{m\infty}^T(k) = \begin{bmatrix} 0.8951 & 0.5653 & 0.8867 & -0.3472 \\ 0.4561 & 1.0210 & -0.5937 & 0.2652 \end{bmatrix}$$

$$W_{c\infty}^T(k) = \begin{bmatrix} -0.1715 & 1.7866 & -1.6280 & 0.8758 \end{bmatrix}$$

$$W_{a\infty}^T(k) = \begin{bmatrix} -0.1878 & 3.6138 & 0.8578 & -1.0491 \\ 0.8590 & 0.5074 & -1.5343 & 0.8021 \end{bmatrix}$$

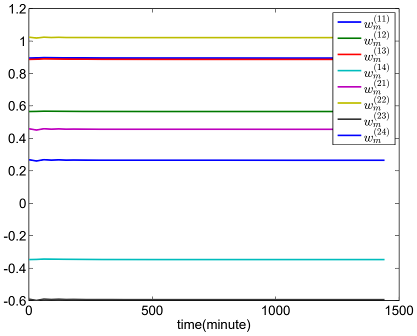


Fig. 8. Weights of model network using Algorithm 2.

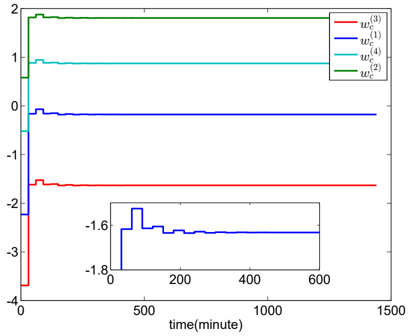


Fig. 9. Weights of critic network using Algorithm 2.

C. Comparison Simulation Experiment Using Standard Policy Iteration

In this section, a comparison simulation experiment is presented. In Algorithm 2, we update the weights of both critic neural network and actor neural network in a novel interleaved manner during the same iteration, whereas in papers [14] and [15], the weights of critic neural network and actor neural network are updated in different iterations. Indeed, in standard Policy Iteration, the weights of the critic NN are updated until they converge, then the weights of the actor NN are updated until they converge. In order to verify the effectiveness of the proposed method, we compare the results of our Algorithm 2 to the results using Policy Iteration (PI).

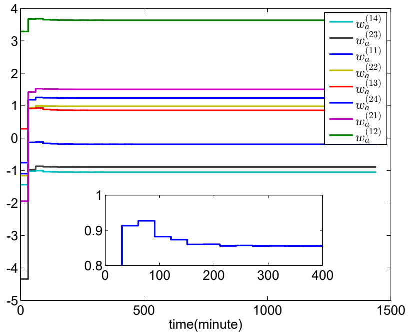


Fig. 10. Weights of neural network using Algorithm 2.

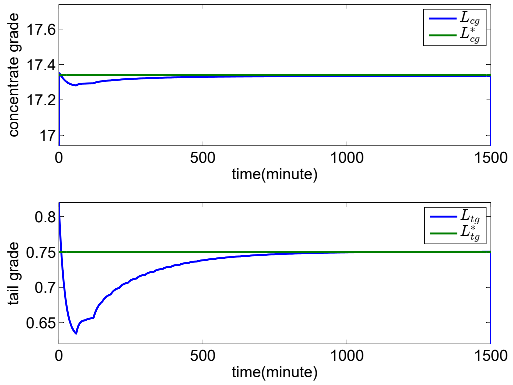


Fig. 11. Operational index tracking results using PI.

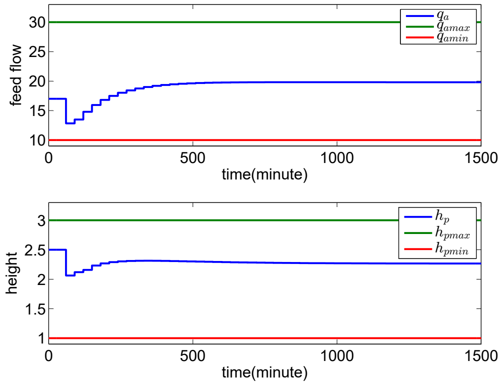


Fig. 12. Optimal control inputs using PI.

Fig. 11 shows the simulation experiment tracking results using PI. The operational indices of the flotation process indeed do track the goal operational indices. However, Fig. 11 shows that the convergence to the setpoints is very slow compared to Fig. 6. Fig. 12 shows the PI simulation inputs, which also satisfy the constraints since the same performance index (3) was used. However, the control magnitudes in Fig. 12 are larger than in Fig. 7.

Fig. 13, Fig. 14 and Fig. 15 show the output-layer weights of the three neural networks. It is seen that the actor NN weights converge slower than interleaved learning in Fig. 10.

D. Comparison Simulation Experiment Using Standard Value Iteration

In this section, a comparison simulation experiment based on VI is presented. Fig. 16 shows the simulation experiment tracking results using VI. The operational indices of the flotation process indeed do track the goal operational indices. Fig. 17 shows the VI simulation inputs, which also satisfy the constraints since the same performance index (3) was used. However, the control magnitudes in Fig. 17 are larger than in Fig. 7. Fig. 13, Fig. 14 and Fig. 15 show the output-layer weights of the three neural networks.

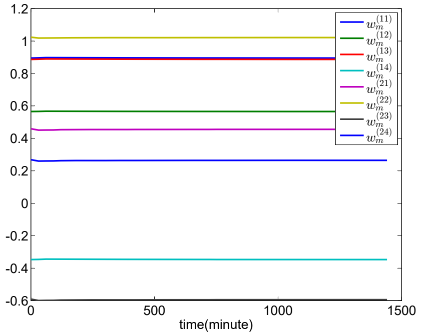


Fig. 13. Weights of model neural network using PI.

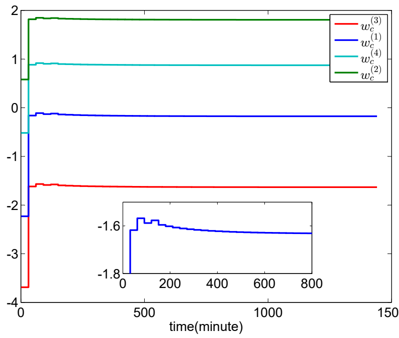


Fig. 14. Weights of critic neural network using PI.

Fig. 18, Fig. 19 and Fig. 20 show the output-layer weights of the three neural networks. It is seen that the actor NN weights converge slower than interleaved learning in Fig. 10.

To evaluate the control performance, the integral absolute error (IAE) and the mean square error (MSE) [49] are used, the evaluation equation is given by

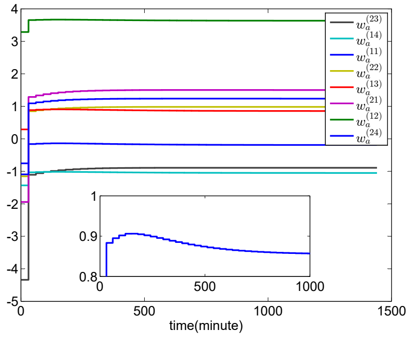


Fig. 15. Weights of actor neural network using PI.

TABLE II
THE EVALUATION INDICES OF ALGORITHM 2, PI AND VI

$2 \leq k^* \leq 49$	IAE_{y_1}	IAE_{y_2}	MSE_{y_1}	MSE_{y_2}
Algorithm 2	0.2779	0.0928	0.0063	0.0058
PI	0.5390	0.6836	0.0154	0.0279
VI	1.3044	2.1080	0.0537	0.0831

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2761852, IEEE Transactions on Industrial Informatics

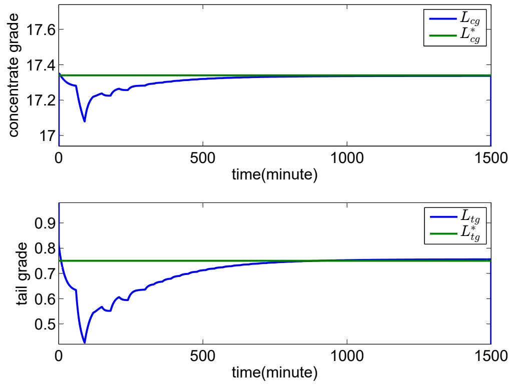


Fig. 16. Operational index tracking results using VI.

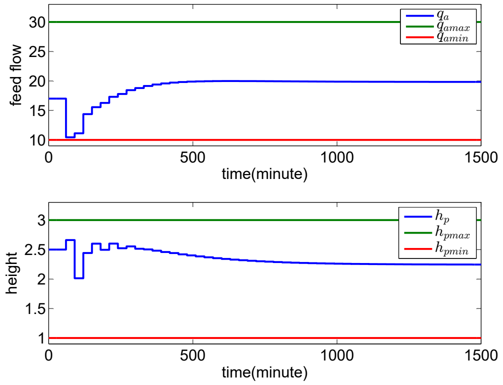


Fig. 17. Optimal control inputs using VI.

$$IAE_{y_i} = \sum_{k=1}^{k^*} |y_i^*(k) - y_i(k)|, \quad i = 1,2 \tag{76}$$

$$MSE_{y_i} = \sqrt{\frac{1}{k^*}\sum_{k=1}^{k^*} \left|y_i^*(k) - y_i(k)\right|^2}, \quad i = 1,2 \tag{77}$$

The evaluation indices of Algorithm 2, PI and VI are shown in Table II. Clearly, the performance on our new Interleaved Learning Algorithm, where the critic NN and actor NN are updated alternately once per iteration, far exceeds that of standard PI and VI, where each NN is iterated to convergence before the other is updated.

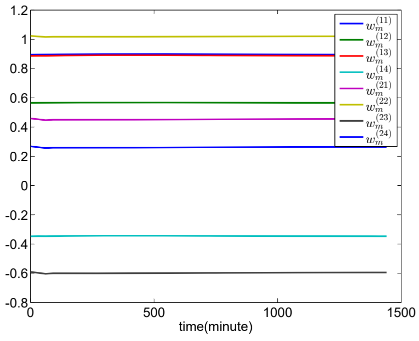


Fig. 18. Weights of model neural network using VI.

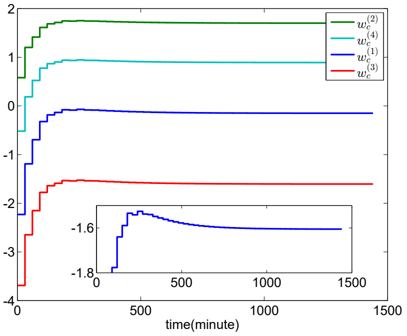


Fig. 19. Weights of critic neural network using VI.

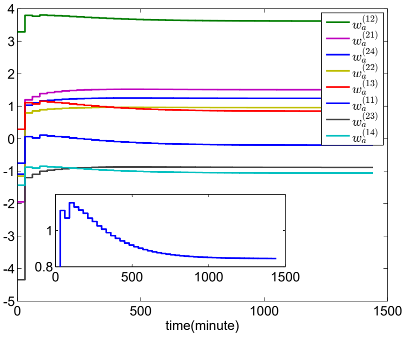


Fig. 20. Weights of actor neural network using VI.

VI. Conclusion

This paper proposes a model-free data-driven operational optimal control based on reinforcement learning applied to single-cell flotation industrial process, which is based on three neural networks, model neural network, critic neural network and actor neural network. This method learns the solution of the tracking problem online without requiring knowledge of the flotation industrial process drift dynamics. The model neural network, critic neural network and actor neural network are updated simultaneously and used previous stored input and output data of the flotation industrial process and value function, instead of their current values, to avoid the requirement of the flotation industrial process model dynamics.

References

[1] T. Chai, S. J. Qin, and H. Wang, "Optimal operational control for complex industrial processes," *Annual Reviews in Contr.*, vol. 38, no. 1, pp. 81–92, Apr. 2014.

[2] T. Chai, J. Ding, and F. Wu, "Hybrid intelligent control for optimal operation of shaft furnace roasting process," *Contr. Eng. Practice*, vol. 19, no. 3, pp. 264–275, Mar. 2011.

[3] Jms-Jounela, M. Dietrich, K. Halmevaara, and O. Tiili, "Control of pulp levels in flotation cells," *Contr. Eng. Practice*, vol. 11, no. 1, pp. 73–81, Jan. 2003.

[4] S. E. Sequeira, M. Graells, , and L. Puigjaner, “Real-time evolution for on-line optimization of continuous processes,” *Ind. Eng. Chem. Res.*, vol. 41, no. 7, pp. 1815–1825, Apr. 2002.

[5] S. Yin, H. Luo, and S. X. Ding, “Real-time implementation of fault-tolerant control systems with performance optimization,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, May 2014.

[6] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Contr. Eng. Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003.

[7] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez, “Model predictive control-a simple and powerful method to control power converters,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1826–1838, Jun. 2009.

[8] F. Liu, H. Gao, J. Qiu, S. Yin, J. Fan, and T. Chai, “Networked multirate output feedback control for setpoints compensation and its application to rougher flotation process,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 460–468, Jan. 2014.

[9] R. Amrit, J. B. Rawlings, and L. T. Biegler, “Optimizing process economics online using model predictive control,” *Comput. & Chem. Eng.*, vol. 58, no. 45, pp. 334–343, Nov. 2013.

[10] C. Muoz and A. Cipriano, “An integrated system for supervision and economic optimal control of mineral processing plants,” *Minerals Eng.*, vol. 12, no. 6, pp. 627–643, Jun. 1999.

[11] T. Chai, L. Zhao, J. Qiu, F. Liu, and F. Jialu, “Integrated network-based model predictive control for setpoints compensation in industrial processes,” *IEEE Trans. Ind. Inf.*, vol. 9, no. 1, pp. 417–426, Feb. 2013.

[12] W. Xue, J. Fan, and Y. Jiang, “Flotation process with model free adaptive control,” in *IEEE Inter. Conf. Inf. Automa.* Macao, CN, Jul. 2017, pp. 442–447.

[13] P. Zhou, T. Chai, and J. Sun, “Intelligence-based supervisory control for optimal operation of a dcs-controlled grinding system,” *IEEE Trans. Contr. Syst. Technol.*, vol. 21, no. 1, pp. 162–175, Jan. 2013.

[14] X. Lu, B. Kiumarsi, T. Chai, and F. L. Lewis, “Data-driven optimal control of operational indices for a class of industrial processes,” *IET Control Theory & Appl.*, vol. 10, no. 12, pp. 1348–1356, Aug. 2016.

[15] W. Dai, T. Chai, and S. X. Yang, “Data-driven optimization control for safety operation of hematite grinding process,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2930–2941, May 2015.

[16] D. Wang, H. He, C. Mu, and D. Liu, “Intelligent critic control with disturbance attenuation for affine dynamics including an application to a microgrid system,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4935–4944, Jun. 2017.

[17] D. Wang, H. He, and D. Liu, “Improving the critic learning for event-based nonlinear h control design,” *IEEE Trans. Cyber.*, 2017, dOI: 10.1109/TCYB.2017.2653800.

[18] D. Wang, D. Liu, Q. Zhang, and D. Zhao, “Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics,” *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 46, no. 11, pp. 1544–1555, Nov. 2016.

[19] F. L. Lewis and D. Liu, “Reinforcement learning and approximate dynamic programming for feedback control,” *IEEE Circuits & Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Feb. 2009.

[20] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control, 3rd Edition*. John Wiley & Sons, Jan. 2012.

[21] R. S. Sutton and A. G. Barto, “Reinforcement learning: An introduction, bradford book,” *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 285–286, Jan. 2005.

[22] D. Liu and Q. Wei, “Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 43, no. 2, pp. 779–789, Apr. 2012.

[23] B. Kiumarsi and F. L. Lewis, “Critic-based optimal tracking for partially unknown nonlinear discrete-time systems,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 1, pp. 140–151, Jan. 2015.

[24] A. Altamimi, F. L. Lewis, and M. Abukhalaf, “Discrete-time nonlinear hjb solution using approximate dynamic programming: convergence proof,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[25] W. Gao, Y. Jiang, Z. P. Jiang, and T. Chai, “Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming,” *Automatica*, vol. 72, pp. 37–45, Oct. 2016.

[26] W. Gao and Z. P. Jiang, “Adaptive dynamic programming and adaptive optimal output regulation of linear systems,” *IEEE Trans. Automat. Contr.*, vol. 61, no. 12, pp. 4164–4169, Dec. 2016.

[27] W. Gao, Z.-P. Jiang, and K. Ozbay, “Data-driven adaptive optimal control of connected vehicles,” *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 5, pp. 1122–1133, May 2017.

[28] F. L. Lewis, A. Yesildirak, and S. Jagannathan, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor & Francis, Inc., 1998.

[29] S. S. Haykin, *Neural networks: a comprehensive foundation*. Tsinghua University Press, Beijing, CN, 2001.

[30] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, “Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming,” *Automatica*, vol. 48, no. 8, pp. 1825–1832, Aug. 2012.

[31] T. Chai, Y. Jia, H. Li, and H. Wang, “An intelligent switching control for a mixed separation thickener process,” *Contr. Eng. Practice*, vol. 57, pp. 61–71, Dec. 2016.

[32] X. Liu, Q. Cheng, J. Li, and X. Zhou, “Integrated automation system for flotation processes,” *Contr. Eng. China*, vol. 23, no. 11, pp. 1702–1706, Nov. 2016.

[33] M. Abu-Khalaf and F. L. Lewis, “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach,” *Automatica*, vol. 41, no. 5, pp. 779–791, May 2005.

[34] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, “Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013.

[35] S. E. Lyshevski, “Optimal control of nonlinear continuous-time systems: design of bounded controllers via generalized nonquadratic functionals,” in *Proc. Amer. Control Conf.* Philadelphia, PA, USA, Jun. 1998, pp. 205–209.

[36] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, “Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems,” *Automatica*, vol. 50, no. 1, pp. 193–202, Jan. 2014.

[37] B. Igelnik and Y.-H. Pao, “Stochastic choice of basis functions in adaptive function approximation and the functional-link net,” *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[38] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, “Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers,” *IEEE Control Syst.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[39] D. Liu and Q. Wei, “Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.

[40] K. G. Vamvoudakis and F. L. Lewis, “Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem,” *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.

[41] H. Xu and S. Jagannathan, “Stochastic optimal controller design for uncertain nonlinear networked control system via neuro dynamic programming,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 24, no. 3, pp. 471–484, Mar. 2013.

[42] A. Sahoo, H. Xu, and S. Jagannathan, “Neural network-based event-triggered state feedback control of nonlinear continuous-time systems,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 27, no. 3, pp. 497–509, Mar. 2016.

[43] Y. Jiang, J. Fan, T. Chai, and T. Chen, “Setpoint dynamic compensation via output feedback control with network induced time delays,” in *Proc. Amer. Contr. Conf.* Chicago, IL, USA, Jul. 2015, pp. 5384–5389.

[44] Z. T. Mathe, M. C. Harris, and C. T. O’Connor, “A review of methods to model the froth phase in non-steady state flotation systems,” *Minerals Eng.*, vol. 13, no. 2, pp. 127–140, Feb. 2000.

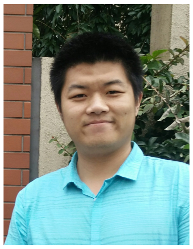
[45] R. Prez-Correa, G. Gonzlez, A. Casali, A. Cipriano, R. Barrera, and E. Zavala, “Dynamic modelling and advanced multivariable control of conventional flotation circuits,” *Minerals Eng.*, vol. 11, no. 11, pp. 333–346, Apr. 1998.

[46] J. Fan, Y. Jiang, and T. Chai, “Operational feedback control of industrial processes in a wireless network environment,” *Acta Automatica Sinica*, vol. 42, no. 8, pp. 1166–1174, Aug. 2016.

[47] Rojas, D. Cipriano, and Aldo, “Model based predictive control of a rougher flotation circuit considering grade estimation in intermediate cells,” *Dyna*, vol. 78, no. 166, pp. 29–37, Apr. 2011.

[48] J. Fan, Y. Jiang, and T. Chai, “Mpc-based setpoint compensation with unreliable wireless communications and constrained operational conditions,” *Neurocomputing*, vol. 270, pp. 110–121, Dec. 2017.

[49] Y. Jia and T. Chai, “A data-driven dual-rate control method for a heat exchanging process,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4158–4168, May 2017.



Yi Jiang (S'14) was born in Hubei Province, China. He received the B. Eng. degree in automation and M.S. degree in control theory and control engineering from information science and engineering collage and State Key Laboratory of Synthetical Automation for Process Industries in Northeastern University, Shenyang, Liaoning, China in 2014 and 2016, respectively, where is currently working toward the Ph.D. degree.

From January to July, 2017, he was a Visiting Scholar with the UTA Research Institute, University of Texas at Arlington, TX, USA. His research interests include networked control systems, industrial process operational control, and reinforcement learning.

Jialu Fan (M'12) received the B.E. degree in automation from Northeastern University, Shenyang, China, in 2006, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2011.

She was a Visiting Scholar with the Pennsylvania State University during 2009-2010. Currently, she is an associate professor with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China. Her research interests include networked control systems, delay-tolerant networks, and mobile social networks.

Tianyou Chai (M'90-SM'97-F'08) received the Ph.D. degree in control theory and engineering from Northeastern University, Shenyang, China, in 1985.

He has been with the Research Center of Automation, Northeastern University, Shenyang, China, since 1985, where he became a Professor in 1988 and a Chair Professor in 2004. He is the founder and Director of the Center of Automation, which became a National Engineering and Technology Research Center in 1997. He has made a number of important contributions in control technologies and applications. He has authored and coauthored two monographs, 84 peer reviewed international journal papers and around 219 international conference papers. He has been invited to deliver more than 20 plenary speeches in international conferences of IFAC and IEEE. His current research interests include adaptive control, intelligent decoupling control, integrated plant control and systems, and the development of control technologies with applications to various industrial processes.

Prof. Chai is a member of the Chinese Academy of Engineering, an academician of International Eurasian Academy of Sciences, and IFAC Fellow. He is a distinguished visiting fellow of The Royal Academy of Engineering (UK) and an Invitation Fellow of Japan Society for the Promotion of Science (JSPS). For his contributions, he has won three prestigious awards of National Science and Technology Progress, the 2002 Technological Science Progress Award from the Ho Leung Ho Lee Foundation, the 2007 Industry Award for Excellence in Transitional Control Research from the IEEE Control Systems Society, and the 2010 Yang Jia-Chi Science and Technology Award from the Chinese Association of Automation.

Jinna Li (M'12) received the M.S. degree and the Ph.D. degree from Northeastern University, Shenyang, China, 2006 and 2009, respectively. She is an associate professor at Shenyang University of Chemical Technology, Shenyang, China.

From April 2009 to April 2011, she was a postdoctor with the Lab of Industrial Control Networks and Systems, Shenyang Institute of Automation, Chinese Academy of Sciences. From June 2014 to June 2015, she was a Visiting Scholar granted by China Scholarship Council with Energy Research Institute, Nanyang Technological Univerisy, Singapore. From September 2015 to June 2016, she was a Domestic Young Core Visiting Scholar granted by Ministry of Education of China with State Key Lab of Synthetical Automation for Process Industries, Northeastern University. From January 2017 until now, she was a visiting scholar with the University of Manchester, the United Kingdom. Her current research interests include operational optimal control, reinforcement learning, distributed optimization control, approximate dynamic programming.

Frank L. Lewis (S'70-M'81-SM'86-F'94) received the bachelors degree in physics/electrical engineering and the M.S.E.E. degree from Rice University, Houston, TX, USA, the M.S. degree in aeronautical engineering from the University of West Florida, Pensacola, FL, USA, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA.

He is currently a Distinguished Scholar Professor and Distinguished Teaching Professor with the University of Texas at Arlington, Fort Worth, TX, USA, the Moncrief-ODonnell Chair of the University of Texas at Arlington Research Institute, Fort Worth, the Qian Ren Thousand Talents Professor and Project 111 Professor with Northeastern University, Shenyang, China, and a Distinguished Visiting Professor with the Nanjing University of Science and Technology, Nanjing, China. He is involved in feedback control, intelligent systems, cooperative control systems, and nonlinear systems. He has authored six U.S. patents, numerous journal special issues, journal papers, and 20 books, including Optimal Control, Aircraft Control, Optimal Estimation, and Robot Manipulator Control, which are used as university textbooks worldwide.

Prof. Lewis is a member of the National Academy of Inventors, a fellow of the International Federation of Automatic Control and the Institute of Measurement and Control, U.K., a Texas Board of Professional Engineer, a U.K. Chartered Engineer, and a founding member of the Board of Governors of the Mediterranean Control Association. He was a recipient of the Fulbright Research Award, the NSF Research Initiation Grant, the Terman Award from the American Society for Engineering Education, the Gabor Award from the International Neural Network Society, the Honeywell Field Engineering Medal from the Institute of Measurement and Control, U.K., the Neural Networks Pioneer Award from IEEE Computational Intelligence Society, the Outstanding Service Award from the Dallas IEEE Section, and the Texas Regents Outstanding Teaching Award in 2013. He was elected as an Engineer of the year by the Fort Worth IEEE Section. He was listed in the Fort Worth Business Press Top 200 Leaders in Manufacturing.