Computers and Chemical Engineering 140 (2020) 106939

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

A hierarchical constrained reinforcement learning for optimization of bitumen recovery rate in a primary separation vessel

Hareem Shafi[a], Kirubakaran Velswamy[a], Fadi Ibrahim[a], Biao Huang[a,*]

Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada

ARTICLE INFO

Article history:
Received 20 February 2020
Revised 17 April 2020
Accepted 20 May 2020
Available online 29 May 2020

Keywords:
Primary separation vessel
Oil sands
Machine learning
Reinforcement learning
Process control

ABSTRACT

This work proposes a two-level hierarchical constrained control structure for reinforcement learning (RL) with application in a Primary Separation Vessel (PSV). The lower level is concerned with servo tracking and regulation of the interface level against variances in ore quality by manipulating middlings flow rate. At the higher level, with the objective to optimize bitumen recovery rate, a supervisory interface level setpoint control is implemented. To prevent sanding, tailings density regulation using tailings withdrawal flow rate is proposed. For each case, an asynchronous advantage actor-critic (A3C) based agent is chosen to interact with a high-fidelity PSV model to learn the near optimal control strategy through episodic interactions. Each of the three control loops is sequentially learnt. In the interface level control loop, a behavioral cloning based two-phase learning scheme to promote stable state space exploration is proposed. The proposed hierarchical structure successfully demonstrates improved bitumen recovery rate by manipulating the interface level while preventing sanding.

1. Introduction

Canada has the third-largest proven oil reserves in the world. These oil reserves exist primarily in the form of oil sands; a loose formation of sand grains or solid sandstone with clay, interspersed with bitumen, a heavy and viscous form of crude oil. Bitumen can be extracted and processed to produce crude oil (Cleveland and Morris, 2014). The Canadian oil sands industry has a capacity of producing 166.3 billion barrels of crude oil products (Government of Canada, 2018). Sales from oil sands producers alone added up to CAD$40 billion in 2016 (CAPP, 2016). The revenue and employment opportunities generated contribute significantly to the national economy.

One-fifth of the total oil sands production is based on open-pit ore extraction. It starts with the mining phase where oil sands ore is shoveled out of the ground. The mined ore is then crushed and transported for the extraction phase. For extraction, heat and chemicals are added to the crushed ore to form a slurry mixture. This mixture is then sent to a gravity separation vessel known as the Primary Separation Vessel (PSV). Once the slurry is fed into the PSV through a feed stream, it forms three distinct layers due to the difference in their densities. These layers are known as the froth layer, the middlings layer, and the tailings layer. The process described is illustrated by means of a block diagram in Fig. 1.

The froth layer that contains mostly bitumen (around 60% bitumen, 10% solids, and 30% water), floats to form the top layer and overflows to upgrading for further treatment. The heaviest particles precipitate at the bottom forming the tailings layer which is withdrawn for further processing before being disposed into a tailings pond. The remaining composition contains mostly water (59% water, 24% bitumen, and 17% solids) and forms the middlings layer in between the froth and the tailings layer. A middlings side stream is pumped from the middle of the vessel to a secondary separation phase for further treatment to recover the leftover bitumen that does not float to the top froth layer.

A highly efficient PSV achieves maximum recovery of bitumen relative to water and solid particles. It reduces the additional processing load on the downstream separation processes. This is owing to the fact that high-quality froth obtained from primary separation requires less processing and energy to remove the remaining solids and water. Hence, the PSV plays a major role in gravity-based separation of bitumen from oil sands. Optimal recovery of bitumen through froth, and overall efficiency of the extraction process, plays a crucial role in the economic and environmental impact that the oil sands industry creates (Gilbert, 2004). Hence, optimal operation of PSV can help achieve environmental and financial targets.

Hence, with the objective of improving the bitumen recovery rate, the first process variable to be considered is the froth-

* Corresponding author.
E-mail addresses: hareem@ualberta.ca (H. Shafi), velswamy@ualberta.ca (K. Velswamy), fibrahim@ualberta.ca (F. Ibrahim), biao.haung@ualberta.ca (B. Huang).

https://doi.org/10.1016/j.compchemeng.2020.106939
0098-1354/

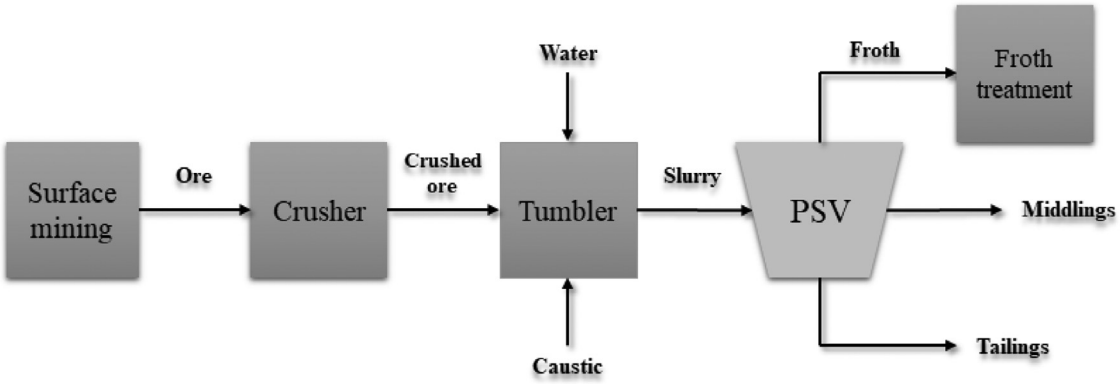

Fig. 1. Block Diagram of the Ore Handling Process.

middlings interface level which directly affects the bitumen recovery rate and has to be regulated within an operational range. Otherwise, it can result in either reducing the quality of the froth being recovered (Li et al., 2011), or losing bitumen to the tailings layer causing further contamination of the tailings pond (Masliyah et al., 1981). Besides improving the bitumen recovery, it is crucial to control the density of the tailings layer which is to be maintained below a certain operational value to avoid excess sand bed build-up in the vessel bottom (Masliyah et al., 1984). This can lead to complete pipeline plugging, also referred to as the 'sanding' phenomenon, particularly if it is associated with a lower tailings withdrawal flow rate.

The theoretical and experimental work reported in Masliyah et al., 1981 provides the foundation for modeling the PSV. This forms the basis for evaluation of the separation performance of PSV in general. More specifically, control oriented as well as operating range oriented studies have also been reported. A typical control problem was developed in Masliyah et al. (1984) with classical multi-loop Proportional Integral (PI) controllers for interface level, and tailings density. However, improving the bitumen recovery was not considered in the control objective.

In Liu et al. (2015), an improved economic model predictive controller (MPC) scheme was applied on a PSV model. The objective was to maximize the overall recovery rate of bitumen without, however, considering the sanding problem. In Gilbert (2004), optimal input trajectories were calculated off-line for different known ore grade transitions. The actual implementation of these optimal trajectories requires the PSV operators to have prior knowledge of the ore quality which limits the applicability of such open-loop control only to known ore grades.

Based on the existing literature, it is concluded that factors such as ore grade, feed flow rate, assumed particle size distribution, and other uncertainties related to modeling assumptions are uncontrollable. They constitute sources of uncertainties and disturbances. Consequently, it impacts the density of the tailings and the middlings layer, resulting in reduced bitumen recovery, and affects separation performance of the PSV in general. None of the existing works actually have taken into consideration the impact of unpredicted nature of all different disturbances on the separation performance. Therefore, we cope with such challenges by using a model free approach like RL in order to provide a generalized solution to such a complex problem. Morever, the recovery rate has direct implication on the economies of this extraction scheme. The secondary extraction via middlings adds up to the cost. To minimize this impact, the control scheme is formulated as a hierarchical structure to optimize the bitumen recovery. This is done through the manipulation of the froth-middlings interface level. Also, the density of the tailings layer is regulated to prevent sanding in the tailings layer.

Reinforcement learning (RL) has gained popularity as a control scheme in recent times due to its ability to learn through trial and error. RL algorithms learn by interacting directly with the environment to sample the optimal actions in order to achieve a specified goal (Sutton and Barto, 2017). In the RL context, the action selection is carried out by the agent, and the process with which the agent interacts is the environment. The framework of their interaction is a Markov decision process (MDP). In a MDP, there are states that the environment can assume, actions that the agent can take, and the reward that is obtained by virtue of taking a particular action in the current observed states. The agent's state to action mapping vector is known as the policy while the cumulative long-term rewards are called the value/action-value function.

RL borrows its formal structure from optimal control where the objective is to design a controller to minimize an objective function of a dynamical system's behavior over time (Sutton et al., 1991). The approach towards solving this problem considers the state to generate actions and then the value function is used to improve the choice of actions for the dynamical system. This is considered to satisfy Bellman optimality. It is from here that the discrete stochastic version of the optimal control structure, MDP, hails from. Employing temporal-difference (TD) learning to find the optimal policy for a MDP in the 1980's resulted in the reinforcement learning structure that is now widely utilized (Sutton and Barto, 2017).

Q-learning, which considers the action-value function (Q-function) in learning the policy was instrumental in the initial popularity of RL (Watkins and Holloway, 2014). It was, however, limited by the curse of dimensionality. A solution to this problem was proposed in the form of neural network based function approximators to estimate the Q-function (DQN) for higher dimensional state spaces in RL problems (Mnih et al., 2015). This enabled control of continuous state space environments with discrete, finite action spaces. Continuous action space optimization was made possible with the introduction of deterministic policy gradient (DPG) algorithm, which employed a neural network approximator for the policy, enabling continuous state to action mapping (Silver et al., 2014). DPG had the ability to control continuous states through continuous actions. However, not only was it computationally expensive, it also suffered from large variance in its gradients. This is attributed to the Monte-Carlo type learning leading to uncorrelated samples. The sample efficiency was further improved in deep deterministic policy gradient (DDPG) which combined DPG and DQN in an actor-critic type model-free architecture for solving more than 20 simulated physics tasks (Lillicrap et al., 2015).

Actor-critic algorithms constitute of an actor which represents the policy and a critic that represents the action-value function. They employ a Monte-Carlo kind of scheme where learning occurs over experience. Experience is gained from multiple repeated episodes to regress an approximation for returns in the form of the action-value function in actor-critic methods (Nguyen et al., 2018). This allows the actor-critic algorithms to be model-free (Sutton and Barto, 2017). In off-policy schemes, the local policy interacts with the environment, from which the rewards and consequently the returns are calculated. Based on the returns, the local policy pulls the global policy to optimize the returns. Every update to the actor is preceded by an update to the critic. The update to the critic is based on minimizing a mean square error (MSE) criterion in predicting the returns, meant to improve the estimate from the action-value function. Using this sequential approach to learning in actor-critic, convergence to a near optimal policy can be guaranteed (Sutton and Barto, 2017).

Continuous space control was demonstrated using DDPG on a variety of 3D tasks (Schulman et al., 2015), a combination of DQN and DDPG for mobile robot control (Tai et al., 2017), stochastic value gradients (SVG) on several physics tasks (Heess et al., 2015), and an asynchronous variant of actor-critic on Atari domain (Mnih et al., 2016). Due to these successful RL implementations in various domains, it makes sense to extend it to process control applications. Drawing analogy between the two, the goal of the RL agent in the process control domain would be to keep a multivariable process within safe operational limits while maintaining it at the setpoint despite process disturbances and measurement noise (Shin et al., 2019).

The ability of RL algorithms to self-learn from direct interaction with the process data make them suitable for use with nonlinear processes where deriving the process model might not be possible or accurate (Spielberg et al., 2017). Due to their self-learning nature, they also have the ability to adapt to process disturbances and shifts in operating conditions. Previously, a successful control of thermostat scheduling for office space in a discrete action space setting has been reported (Wang et al., 2017). Also, continuous space optimization using a policy gradient based approach has also been reported (Wang et al., 2018). These schemes were based on on-policy proximal actor-critic setup.

A model-based RL method is used with deep neural network (DNN) approximators to address the finite horizon optimal control problem in Kim et al., 2020. The control problem is formulated in Hamiltonian-Jacobi-Bellman (HJB) format with illustration on a nonlinear batch reactor and 1-dimensional diffusion-convection-reaction process. The authors in Lee, Jong Min and Lee, Jay H. (2005) propose and contrast two approximate dynamic programming approaches using function approximation, a model-based approach and model-free Q-learning for data-driven control of nonlinear processes implemented on a CSTR. While these two papers address continuous tracking control using DNNs and a nearest neighbour local averager, they are not concerned with production or economic optimization. A factorial policy based RL solution for production optimization of large-scale chemical plant was presented in Cui, Yunduan and Zhu, Lingwei and Fujisaki, Morihiro and Kanokogi, Hiroaki and Matsubara, Takamitsu (2018). The optimization was carried out using model-free RL and implemented on a vinyl acetate monomer (VAM) plant to maximize the VAM yield and quality while maintaining plant stability. Actor-critic was employed in Ge, Yulei and Li, Shurong and Chang, Peng (2018) to find the optimal Alkali-Surfactant-Polymer injection control strategy to enhance oil recovery taking the net present value (NPV) as the initial performance index. The work in Cui, Yunduan and Zhu, Lingwei and Fujisaki, Morihiro and Kanokogi, Hiroaki and Matsubara, Takamitsu (2018) and Ge, Yulei and Li, Shurong and Chang, Peng (2018) uses RL for optimal control, but does not take into account tracking control and relies on conventional controllers for that.

Nian et al. employed contextual bandit for fault detection and DQN for fault tolerant control of a Wood Berry distillation column (Nian et al., 2019). These works concentrate on tracking control, optimal control or fault tolerant control individually. This work aims to provide a comprehensive solution that tackles both tracking control and optimal control. It proposes an asynchronous advantage actor-critic (A3C) based solution for the optimal control and tracking of the PSV as motivated previously. The contribution of this paper comes from the proposed hierarchical control scheme that tackles the multiple objectives of the PSV. The hierarchical, cascade type structure is proposed for improving the bitumen recovery rate through froth-middlings interface level tracking while regulating the tailings density to prevent sanding. As a novel approach for interface tracking, a semi-supervised scheme based on behavioural cloning is employed during training for safe exploration of the action space.

The rest of the paper is arranged as follows: Section 2 details the high fidelity PSV model used, Section 3 discusses the multi-loop RL control architecture and the experimental setup, Section 4 shares the results and discussions, and finally Section 5 highlights the main conclusions and sets directions for future work.

2. PSV Process model

2.1. Process description

Mass balance with gravity separation principles used in the PSV model in this work are all based on the work in Gilbert (2004) and its references. The gravity separation principles employed and the models presented in this section are taken from Masliyah et al., 1981. The following main assumptions are considered in the model development. The materials' species present in the PSV are bitumen, solids, and water (labelled by the subscript j that takes b, s, w respectively). They are considered to be present in three constant sizes: small, medium, and large. The bitumen and coarse solid particles are assumed to be spherical; whereas, the fine solids follow platelets' shape. The density of species (bitumen, water, and solids particles) are all assumed to be constant and the viscosity of the middlings layer is assumed to be that of water.

Each layer is assumed to be perfectly mixed, contains continuous medium, and modelled using mass balance principles with interactions between layers through froth-middlings and middlings-tailings interfaces. This interaction is characterized by considering particles' movement between layers to follow steady-state settling relationships (Stokes' law). It will be briefly revisited in the forthcoming subsections in combination with hindered settling models for suspension of particles following the reference Concha and Almendra (1979). The froth-middlings interface is considered to be mobile and particles can move back and forth through it. While, the middlings-tailings interface is static and particles only move in a downward direction. The downwards direction is considered to be the positive direction of particles movement along with one-dimensional assumed flow.

In the mass balance equations presented hereafter, no material generation is assumed and can be expressed as expounded in the following subsections. The notation used is presented in Table 1.

2.2. Froth layer

The volume of the froth layer V_f is assumed to be a function of the interface velocity v_I. This is because the top of the froth layer is assumed to be fixed and matches the top of the PSV. It is described by Eq. (1) where A_{vessel} represents the vessel cross-sectional area.

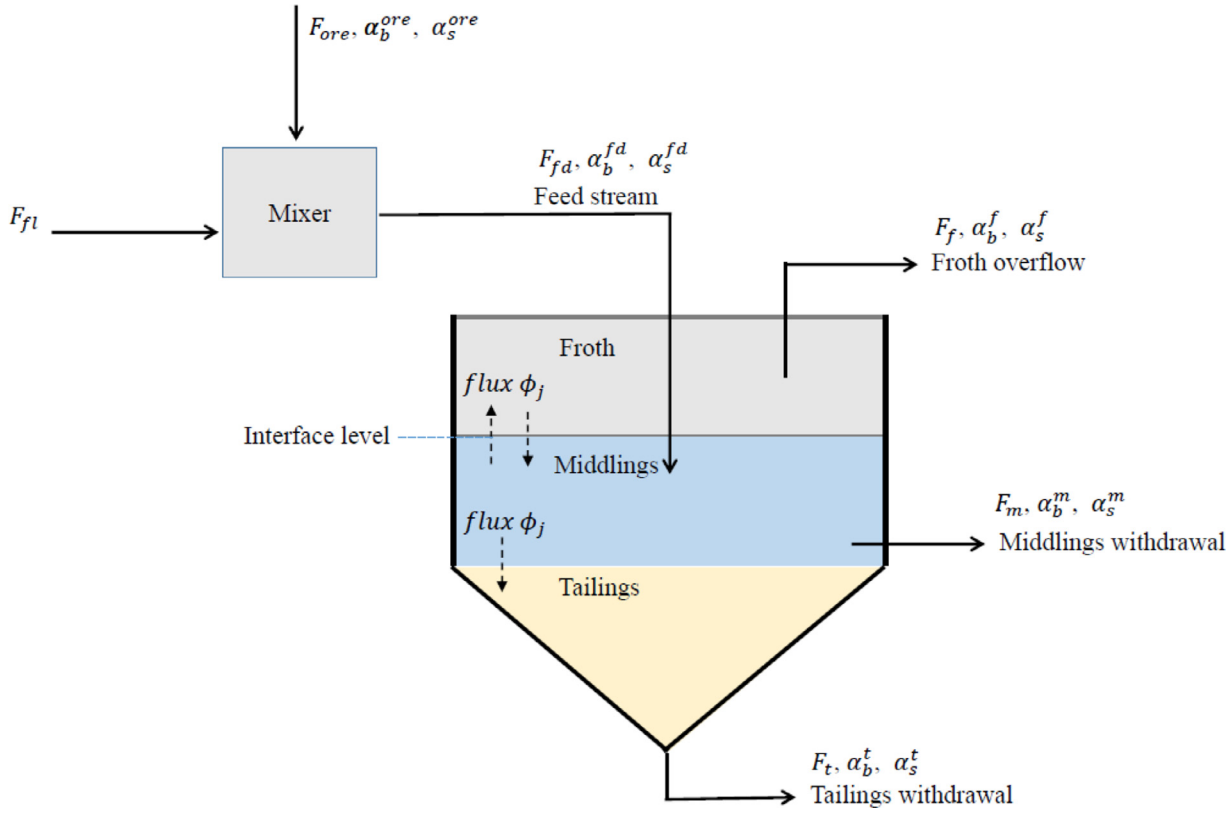

Fig. 2. PSV Schematic.

Table 1
Primary Separation Vessel Model Notation.

Parameter	Description [unit]
$i \in f, m, t$	froth, middlings, and tailings
$j \in b, s, w$	bitumen, sand, and water
$k \in 1, 2, 3$	small, medium, and large particle size
I_{F-M}	froth-middlings interface level [m]
F_{fd}	feed flow rate $[m^3 s^{-1}]$
F_{fl}	flood water flow rate $[m^3 s^{-1}]$
F_f	froth overflow $[m^3 s^{-1}]$
F_m	middlings withdrawal flow rate $[m^3 s^{-1}]$
$\overline{F_m}$	nominal middlings withdrawal flow rate $[m^3 s^{-1}]$
F_t	tailings withdrawal flow rate $[m^3 s^{-1}]$
$\overline{F_t}$	nominal tailings withdrawal flow rate $[m^3 s^{-1}]$
V_i	volume of i^{th} layer $[m^3]$
$V_{i_{sp}}$	volume setpoint of i^{th} layer $[m^3]$
α_j^i	volume fraction of species j in layer i
ϕ_j	flux of species j $[m^3 s^{-1}]$
A_{vessel}	vessel cross sectional area $[m^2]$
v_I	froth-middlings interface velocity $[ms^{-1}]$
v_j^i	settling velocity of species j in layer i $[ms^{-1}]$
ρ_j	density of species j $[kgm^{-3}]$
ρ_i	density of layer i $[kgm^{-3}]$
d_j^k	particle diameter size k of species j $[m]$
g	gravitational constant $[m^2 s^{-1}]$
e_t	error term $[m]$
Θ, ϵ	settling velocity correction factors
η	dynamic viscosity $[kgm^{-1}]$

$$\frac{dV_f}{dt} = A_{vessel} v_I \tag{1}$$

As shown in Fig. 2, a species j's transport occurs as 1) a flux ϕ_j through the interface with the middlings (equation (3)) and 2) leaves the top of the PSV with a flow rate of F_f. Applying mass balance principles, the volumetric fraction of a species j in the froth layer (α_j^f) is described in Eq. (2).

$$\frac{d\alpha_j^f}{dt} = \frac{1}{V_f}(\phi_j - \alpha_j^f F_f - \alpha_j^f A_{vessel} v_I) \tag{2}$$

$$\phi_j = \begin{cases} \alpha_j^m A_{vessel}(v_I - v_j^m), & v_I > v_j^m \\ \alpha_j^f A_{vessel}(v_I - v_j^m), & v_I \le v_j^m \end{cases} \tag{3}$$

where $j \in b, s$. α_j^m is the volumetric fraction of a species j in the middlings layer entering the froth layer. This occurs when the interface velocity v_I is greater than the settling velocity v_j^m of species j in the middlings layer.

As indicated by Gilbert (2004) and the reference within, the settling velocity v_j^m is calculated by equation (4). This equation corrects the free settling velocity v_j^{free} by Concha's correlation (Concha and Almendra, 1979). This correction is considered in order to account for the suspension resulting from the presence of other particles in a layer, so the settling of a particle is hindered as indicated in Eq. (4):

$$v_j^m = v_j^{free} \frac{(1 - 1.45 \sum \alpha^{particles})^{1.83}}{1 + 0.75^{\frac{1}{3}}} \tag{4}$$

The free settling velocity itself v_j^{free} was developed by Swanson (1967) and Swanson (1975) and is based on Stokes' equations for free-settling as shown in Eq. (5):

$$v_j^{free} = \frac{\frac{4}{3} g d^2 (\rho_j - \rho_i)}{\theta_j (2d^{\frac{3}{2}} (\frac{g\rho_j \rho_i}{3})^{\frac{1}{2}} + \sqrt{48}\epsilon_j \eta)} \tag{5}$$

where the shape factors of a species $j \in b, s$, is represented by the parameters θ_j and ϵ_j. g is the gravitational constant and η is the viscosity of water, and d refer to the particle diameter of a species. Three particlesâ sizes are considered for bitumen and three for

sand particles as indicated previously. The shape factor is assumed to be spherical for bitumen and for coarse solid particles and assumed to be platelets for the fine particles (clays).

The suspension in a layer is assumed to be uniform and its density is calculated as the weighted summation of speciesâ densities in it as represented in Eq. (6):

$$\rho_i = \rho_w \alpha_w^i + \rho_b \alpha_b^i + \rho_s \alpha_s^i \tag{6}$$

where $i \in f, m, t$ denotes froth, middlings, and tailings respectively. The subscripts w, b and s indicate the species, namely water, bitumen, and sands respectively. ρ_j indicates the density of a species j either bitumen or sand particle.

The interface velocity v_I is modelled as the Wallis shock-wave equation (Eq. (7)) for a first order approximation as follows (Wallis, 1969):

$$v_I = \frac{\sum_{k=1}^{3} \alpha_{bk}^m v_{bk}^m - \sum_{k=1}^{3} \alpha_{bk}^f v_{bk}^f}{\sum_{k=1}^{3} \alpha_{bk}^m - \sum_{k=1}^{3} \alpha_{bk}^f} \tag{7}$$

where k is the index of particle size (3 sizes were considered) and again α_j^i is the volume fraction of species j in layer i.

2.3. Middlings layer

Similar to the volume of the froth layer, the middlings layer volume V_m is assumed to be only a function of the interface velocity v_I as the middlings-tailings interface is stationary and only the froth-middlings interface is mobile. Thus, the middlings layer is represented as in Eq. (8).

$$\frac{dV_m}{dt} = A_{vessel} v_I \tag{8}$$

As shown in Fig. 2, a species j's transport in the middlings layer occurs as a 1) flux ϕ_j through the interface with both, the froth layer and the tailings layer as indicated in Eq. (10), 2) feed injected slurry with flow rate F_{fd}, and 3) as a withdrawal that leaves the middlings layer with the withdrawal flow rate F_m. Consequently, using mass balance principles, the volumetric fraction of a species $j \in b, s$ in the middlings layer (α_j^m) is described as given in Eq. (9).

$$\frac{d\alpha_j^m}{dt} = \frac{1}{V_m}(\alpha_j^{fd} F_{fd} - \alpha_j^m F_m - \alpha_j^m A_{vessel} v_j^t + \alpha_j^m A_{vessel} v_I + \phi_j) \tag{9}$$

$$\phi_j = \begin{cases} -\alpha_j^m A_{vessel}(v_I - v_j^m), & v_I > v_j^m \\ -\alpha_j^f A_{vessel}(v_I - v_j^m), & v_I \le v_j^m \end{cases} \tag{10}$$

where α_j^{fd} is the volumetric fraction of species j in the feed stream, and v_j^t is the hindered settling velocity of a particle of species j in the tailings layer also calculated using Eq. (4).

2.4. Tailings layer

The volume of the tailings layer is constant as the middlings-tailings interface is considered stationary and this simplifies the model equations. As shown in Fig. 2, a species j transport occurs as a 1) flux ϕ_j through the middlings-tailings interface and 2) as a withdrawal that leaves with the withdrawal flow rate F_t from the bottom of the PSV. The volumetric fraction of species $j \in b, s$ in the tailings layer (α_j^t) is then described as given in Eq. (11).

$$\frac{d\alpha_j^t}{dt} = \frac{1}{V_t}(\alpha_j^m A_{vessel} v_j^t - F_t \alpha_j^t) \tag{11}$$

2.5. Feed equation

As shown in Fig. 2, with a flow rate of F_{ore}, ore is fed to a mixer of volume V_{mix} to be first mixed with flood water of flow rate F_{fl} before being fed into the PSV. Thus, the volumetric fraction of species j in the feed stream (α_j^{fd}) is described in Eq. (12).

$$\frac{d\alpha_j^{fd}}{dt} = \frac{1}{V_{mix}}(\alpha_j^{ore} F_{ore} - \alpha_j^{fd}(F_{ore} + F_{fl})) \tag{12}$$

The following flow rate balance is considered to calculate and constrain the overflow stream:

$$F_{fd} = F_{fl} + F_{ore}$$

$$F_f = F_{fd} - F_m - F_t$$

such that:

$$F_f \ge 0.$$

2.6. Recovery rate

The efficacy of the PSV in extracting a bitumen rich froth directly affects the economic impact of the oil sands industry by determining the load on the downstream processes. This effectiveness is represented by the bitumen recovery rate RR. It depends on the bitumen content in the froth α_b^f and ore α_b^{fd} and the corresponding froth overflow rate F_f and ore flow rate F_{ore} as represented in Eq. (13).

$$RR = \frac{\sum \alpha_b^f F_f}{\sum \alpha_b^{ore} F_{ore}} \tag{13}$$

3. RL based control

3.1. Markov decision process

As previously motivated, the RL framework comprises of: a RL agent that is the learner in the process (analogous to the controller) and the environment (analogous to the plant including the rewarding mechanism). The agent interacts with the environment to optimize a certain facet of its behavior skewed by the designer's choice of reward. This framework is represented by a MDP that follows the Markov property (Littman et al., 2013). It assumes that the present state of the environment is sufficient to make the optimal decision, i.e. it contains the relevant historical information. The MDP encapsulates the agent-environment interaction in discrete time steps within the finite time learning episode $t \in N$. The terminal state of an infinite horizon optimization RL setup is based on an episodic approach. In the following subsections, the projection of PSV's state space into the action space (middlings flow rate F_m, froth-middlings interface level setpoint $I_{F-M_{SP}}$, tailings flow rate F_t) based on the rewarding mechanism is described using this MDP structure.

The specific case of lower level froth-middlings interface level control is used as an example to understand MDP in this subsection. It is assumed that at each time instant t, there is a set of observable states $s_t \in S$ that the environment can assume, such as $s_t = [I_{F-M}, I_{F-M_{SP}}, F_m]$. There is also a set of actions $a_t \in A$ the agent can choose from, given the state observation s_t, to manipulate the F_m to track the interface level. This is done in accordance with its current policy $\pi(a_t|s_t)$. By virtue of the action a_t, the PSV transitions to a new state s_{t+1} and emits a scalar reward r_t associated with being in the new state and the action that had been taken, such as in Eq. (14). The rewards are accumulated over time by following the policy π. They are then corrected by a discount

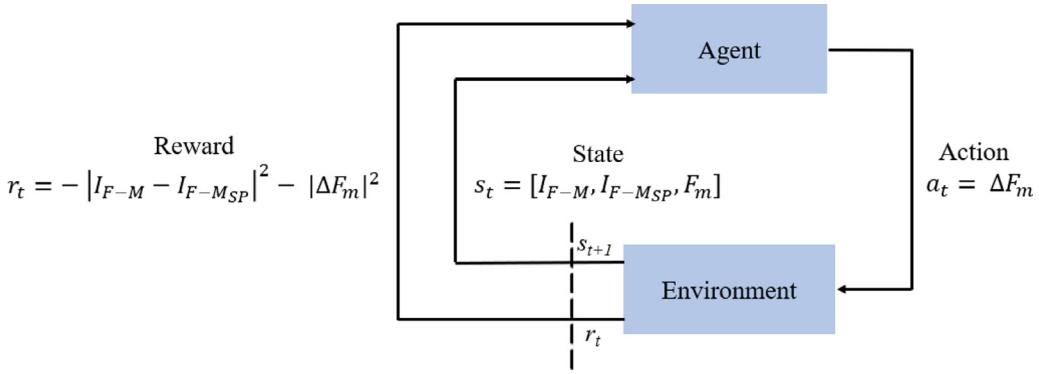

Fig. 3. Markov Decision Process Representation for Lower Level Interface Tracking.

factor γ which helps to keep the returns bounded. This determines the relative importance of future rewards and is represented in the form of returns R_t as given in Eq. (15). Returns in this context are a direct feedback on the agent's performance at each state with reference to the terminal goal. They are calculated by considering the deviation from the setpoint over time. Their estimation, given only the states, is also known as the value function $V(s)$ as shown in Eq. (16). If the action taken (the flow rates chosen) is also considered in obtaining the expectation of returns, they constitute the action-value function $Q(s, a)$ given in Eq. (17). MDP setup for the lower loop corresponding to interface level control in the hierarchical architecture is illustrated in Fig. 3.

$$r_t = - \mid I_{F-M}(t) - I_{F-M_{SP}}(t) \mid^2 - \mid \Delta F_m \mid^2 \tag{14}$$

$$R_t = \sum_{k=0}^{n} \gamma^k (- \mid I_{F-M}(t+k) - I_{F-M_{SP}}(t+k) \mid^2 - \mid \Delta F_m \mid^2) \tag{15}$$

$$V(s) = E_\pi [R_t \mid s] \tag{16}$$

$$Q(s, a) = R_t + \gamma * V(s) \tag{17}$$

3.2. Actor-Critic

Actor-Critic combines the benefits of DQN and DPG (Shin et al., 2019) to allow the state and action sets in the MDP context to transcend from discrete to continuous state and action spaces. Neural networks are employed as function approximators to implement the policy $\pi_{\theta_A}(a_t \mid s_t)$ represented by the actor, and the value function $V_{\theta_C}(s_t)$ represented by the critic (which is monotonic), where θ_A, θ_C represent the neural network parameters for the actor and critic respectively. The objective of actor-critic is to improve the accuracy in estimating the returns using critic, followed by optimizing the estimated returns by updating the actor (Eq. (18)). The learning gradient of the policy is considered an approximate solution to the Bellman optimality equation (Eq. (19)). A baseline term limits the variance in the gradients of the neural network approximators aiding in convergence (Eq. (20)) (Lillicrap et al., 2015).

$$max_{\theta_A} J(\theta_A) = E(R_t \mid \pi_{\theta_A}) \tag{18}$$

$$\nabla_{\theta_A} J(\theta_A) = E_\pi [\sum_{t=0}^{N} \nabla_{\theta_A} \log \pi_{\theta_A}(a_t \mid s_t)[R_t]] \tag{19}$$

$$A(s_t) = R_t - V(s_t) \tag{20}$$

A set of λ tuples containing the state, action, action-value and the reward are recorded in the experience replay buffer for each sample time t until the buffer is full. The experience replay buffer holds the information required to calculate the gradient from losses. Since the objective of the critic is accurate estimation of the returns, the critic parameters are updated by means of the critic loss function as shown in Eq. (21). The returns R_t are calculated from the rewards stored in the experience replay buffer, while the returns estimate (the action-value function) is obtained by passing the state/action information to the critic network.

$$\min_{\theta_C} J(\theta_C) = \sum_{t=0}^{N} || R_t - V_{\theta_C}(s_t) ||^2 \tag{21}$$

After the network parameters' update in the critic network, the actor is updated by means of the actor network gradient derived from its loss. The actor loss is adjusted by the advantage function to reduce variance such as in Advantage Actor-Critic (A2C) where the critic action-value replaces $x(s_t, a_t)$ with $A(s_t, a_t)$ calculation as presented in Eq. (22).

$$\nabla_{\theta_A} J(\theta_A) = E_\pi \left[\sum_{t=0}^{N} \nabla_{\theta_A} \log \pi_{\theta_A}(a_t \mid s_t) \left[A_{\theta_C}(s_t) \right] \right] \tag{22}$$

3.3. Exploration

Policy π can either be deterministic (Eq. (23)), that is, the policy maps the state observations s_t directly to the actions a_t, or stochastic (Eq. (24)), where the policy samples a probability distribution described by μ_t, σ_t from which the action is sampled (Eq. (25)). Stochastic policies inherently promote exploration making it suitable for improved convergence for continuous space, nonlinear chemical processes. Whereas, in the case of deterministic policies, exploration is encouraged by means of schemes such as ϵ-greedy or ϵ-soft.

$$a_t = \pi_{\theta_A}(s_t) \tag{23}$$

$$\mu_t, \sigma_t = \pi_{\theta_A}(s_t) \tag{24}$$

$$a_t \sim N(\mu_t, \sigma_t) \tag{25}$$

The theme of exploration and exploitation is central to reinforcement learning. Exploitation is when the agent chooses the action known to result in the highest returns, while exploration is the agent taking an equal probability action to explore the action space. Furthermore, Shannon's entropy $H(\pi)$ is introduced in the actor loss calculation to encourage exploration in the stochashtic format (Eq. (26)). Higher entropy may result in delayed convergence while preventing convergence to a local optima. The actor

Algorithm Asynchronous Advantage Actor-Critic

- **Output:** Optimal policy $\pi_{\theta_A^*}$
- Initialize global actor parameters θ_A, critic parameters θ_C, counter $T = 0$
- Initialize workers $i \in \{1:n\}$ with worker-specific parameters θ_A^i, θ_C^i
- **For** episodes $j < N$**:**
  - **For** worker i:
    - **Repeat{**
      - Initialize gradients: $d\theta_A \leftarrow 0, d\theta_C \leftarrow 0$
      - Pull worker parameters $\theta_A^i = \theta_A,\ \theta_C^i = \theta_C$
      - $t_{start} = t$
      - Obtain state s_t
      - **Repeat{**
        a. Obtain $a_t \sim N(\mu_t, \sigma_t)$ from $\pi(\mu_t, \sigma_t | s_t, \theta_A^i)$
        b. Implement a_t
        c. Record reward r_t
        d. Record new state s_{t+1}
        e. $t \leftarrow t + 1$
        f. $T \leftarrow T + 1$**}**
      - **Until** $t - t_{start} == t_{max}$
      - **If** terminal s_t: $R = 0$
      - **Else:** $R = V(s_t, \theta_C^i)$
      - **For** $j \in \{t-1, \ldots, t_{start}\}$ **do{**
        a. $R \leftarrow r_j + \gamma R$
        b. Sum gradients wrt θ_C^i: $d\theta_C \leftarrow d\theta_C + {\partial (R - V(s_t; \theta_C^i))^2}/{\partial \theta_C^i}$
        c. Sum gradients wrt θ_A^i: $d\theta_A \leftarrow d\theta_A + \nabla_{\theta_A^i} \log \pi(a_t | s_t; \theta_A^i)(R - V(s_t; \theta_C^i))$**}**
      - **End**
      - Perform asynchronous update of θ_A using $d\theta_A$ and θ_C using $d\theta_C$
    - **Until** $T < T_{max}$
    - **End**
- **Return** $\pi_{\theta_A^*}$

Fig. 4. Pseudocode for A3C adapted from Mnih et al. (2016) for the PSV.

loss is hence represented as given in Eq. (27) where β is a hyperparameter representing the tradeoff between optimizing the advantage function and exploration (Fortunato et al., 2017).

$$H(\pi) = -\sum_t P(a_t) \log P(a_t) \tag{26}$$

$$\nabla_{\theta_A} J(\theta_A) = E_\pi[\sum_{t=0}^{N} \nabla_{\theta_A}[\log \pi_{\theta_A}(a_t \mid s_t)[A_{\theta_C}(s_t)]] - \beta * H(\pi) \tag{27}$$

3.4. Asynchronous advantage actor-critic

The learning approach differs slightly between on-policy algorithms and off-policy algorithms. On-policy algorithms interact with the environment using the same policy that they update to converge towards the optimal policy. Off-policy algorithms interact with the environment using a behavior policy, while a separate target policy is updated to find the optimal policy. Asynchronous advantage actor-critic (A3C) is an instance of such off-policy scheme where a global actor-critic network is updated using the experience gained through multiple local actor-critic behavior policies working asynchronously. Each local actor-critic interacts with its own local copy of the environment (in this case the PSV and the rewarding mechanism) to gain the experience. This aids exploration in the state/action space which is essential for development of a generalized solution for nonlinear process control applications. By employing multiple local copies of actor-critic and its corresponding environment, A3C redistributes the learning between multiple workers by making use of parallel computing. This also leads to improved and stable convergence (Mnih et al., 2016). The pseudocode of the A3C adopted from Mnih et al. (2016) for the PSV is given in Fig. 4. The sequence repeats itself for each worker for each learning episode except for the first time in which each worker interacts with the environment, and no updates are made to the worker networks. As seen in Fig. 4, the A3C scheme would have higher degree of exploration, so the global policy is generalized owing to the asynchronous learning. Hence, the near optimal global policy is assumed to be available at the end of the stipulated episodes.

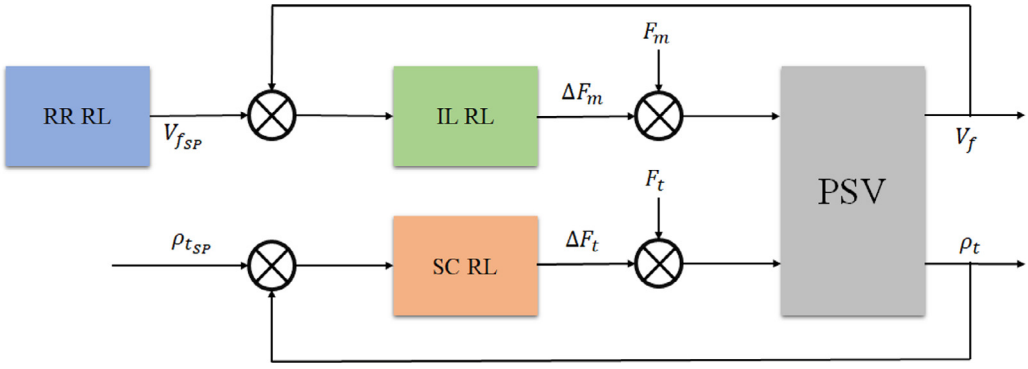

Fig. 5. Multiloop Control of PSV.

3.5. Hierarchical multiloop control

The supervisory layer of the hierarchical agent overlooks optimization of the bitumen recovery rate (RR). The *RR* is optimized through changes in the froth-middlings interface level setpoint $I_{F-M_{SP}}$. A lower level RL agent manipulates the interface level I_{F-M} to track the setpoint changes provided by this agent. In addition, to ensure safe operation of the PSV, an RL agent regulates the tailings density ρ_t below a set threshold to prevent sanding in the tailings. This would otherwise lead to depletion of the middlings or froth layer, leading to poor or no recovery. The control structure is illustrated in Fig. 5. The reward mechanism of each RL agent, the states to the actor and the critic, and the loss functions determine the learning of each agent and comprise the setup. The setup overlooking each objective is explained in Subsections 3.5.1 to 3.5.3.

3.5.1. Low level RL - Interface level control

Control of the froth-middlings interface level is achieved through manipulation of middlings flow rate F_m. The interface level I_{F-M} depends directly on the froth volume V_f as given in equation (28), where A_{vessel} is the area of the vessel. A finite difference type simulation with a sample time of 1 h (with one minute iterations in the inner loop) is executed. A new action ΔF_m is chosen by the actor based on the state observations $s_t^{IL_A}$ every 1 h. The states observed by the actor and the critic, that is the input vector, are given in Eq. (29). Since a stochastic policy is followed, the output of the actor is an action probability distribution as shown in Eq. (30). The normalized action is sampled from the distribution and scaled to the PSV's practical operating range, shown in Eq. (31). The action then updates the middlings flow rate F_m as shown in Eq. (32), where F_{m_s} represents the steady state middlings flow rate.

$$I_{F-M}(t) = \frac{V_f(t)}{A_{vessel}} \tag{28}$$

$$s_t^{IL} = [I_{F-M}(t), I_{F-M_{SP}}(t), F_m(t)] \tag{29}$$

$$\mu_t^{IL}, \sigma_t^{IL} = \pi_{\theta_{IL}}(s_t^{IL}) \tag{30}$$

$$\Delta F_m(t) \sim N(\mu_t^{IL}, \sigma_t^{IL}) \tag{31}$$

$$F_m(t) = F_{m_s} + \Delta F_m(t) \tag{32}$$

The output of the critic estimates returns R_t^{IL} in the form of the value function $V_{\theta_C}(s_t)$ (Eq. (33)).

$$\hat{R_t^{IL}} = V_{\theta_{IL}}(s_t^{IL}) \tag{33}$$

The reward function, given in Eq. (34), is shaped to achieve the multiple control objectives. The first term included intends to minimize the deviation of the interface level from the setpoint (handled in terms of the froth-middlings interface level deviation given in Eq. (35)). The second focuses on minimizing the controller effort. The third term F_m^{breach} is a soft constraint for maintaining the action within operational bounds through penalization as presented in Eq. (36). It ensures that through the course of RL learning, it would learn not to breach the upper/lower bound in order to optimize the rewards.

$$r_t^{IL} = -\mid \Delta I_{F-M}(t)) \mid^2 - \mid \Delta F_m(t) \mid^2 - F_m^{breach} \tag{34}$$

$$\Delta I_{F-M}(t) = I_{F-M}(t) - I_{F-M_{SP}}(t) \tag{35}$$

$$F_m^{breach} = \begin{cases} 0, & if(0.8F_{m_s} \le F_m(t) \le 1.2F_{m_s}) \\ \mid F_m - 1.2F_{m_s} \mid, & if(F_m > 1.2F_{m_s}) \\ \mid F_m - 0.8F_{m_s} \mid, & if(F_m < 0.8F_{m_s}) \end{cases} \tag{36}$$

The critic loss, given in Eq. (21), employs the value function $(V_{\theta_{IL}}(s_t^{IL}))$. Actual returns are calculated from the rewards obtained from Eq. (34). The actor loss is calculated with the advantage function values from the updated critic as shown in equation (22). The results obtained are shared in Section 4.

3.5.2. Supervisory RL - Recovery rate optimization

The recovery rate (*RR*) depends on the bitumen content in the froth (α_b^f) and ore (α_b^{fd}), and the corresponding froth overflow rate (F_f) and ore flow rate (F_{ore}) as given in Eq. (13). Since the bitumen content in the ore and the ore flow rate are beyond control, the froth-middlings interface level (I_{F-M}) is considered to address recovery rate.

The sampling time considered to update the interface level setpoint I_{F-M} is 2 hours. This is in adherence to industrial practice since that is the frequency at which the ore quality measurements from the lab will be available. The input state vector is provided to the actor and the critic every 2 hours, and it is given in Eq. (37). These states are the recovery rate at the given time *RR*, the baseline recovery rate RR_{nom} taken from Liu et al. (2015), and the middlings flow rate F_m. A stochastic policy is followed again, so the output of the actor is an action probability distribution, as shown in Eq. (38). The normalized control action, change in froth-middlings interface level setpoint $\Delta I_{F-M_{SP}}$, is sampled from the given distribution (Eq. (39)) and scaled to a practical operating range (Eq. (40)), that is $\pm 1.2m$. The range for setpoints for froth-middlings interface level is also set between the operating limits of $18.8m$ to $28.2m$.

$$s_t^{RR} = [RR, RR_{nom}, F_m(t)] \tag{37}$$

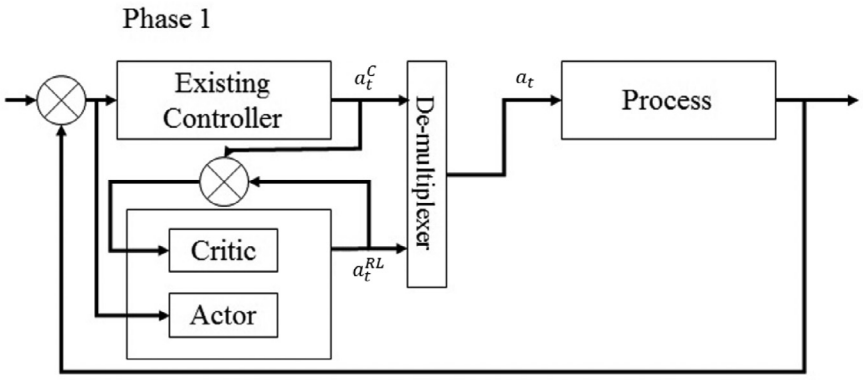

Fig. 6. Phase 1 of Coerced Learning.

$$\mu_t^{RR}, \sigma_t^{RR} = \pi_{\theta_{RR}}(s_t^{RR}) \tag{38}$$

$$\Delta I_{F-M_{SP}}(t) \sim N(\mu_t^{RR}, \sigma_t^{RR}) \tag{39}$$

$$-1.2 \leq \Delta I_{F-M_{SP}}(t) \leq 1.2 \tag{40}$$

The instantaneous rewards reflect the objective to maximize the recovery rate and maintain the system's stability by minimizing the magnitude of the action taken. The reward reflected is positive in the case when the recovery rate is above the nominal recovery rate RR_{SP} and negative otherwise, as given in equation (41). They are used in the actual returns R_t^{RR} calculation.

$$r_t^{RR} = \mid \Delta RR - RR_{nom} \mid^2 - \mid \Delta I_{F-M_{SP}} \mid^2 \tag{41}$$

$$\Delta RR = RR - RR_{nom}$$

The critic loss (Eq. (21)) and actor loss (Eq. (22)) extract information from the supervisory RL agent in a similar manner as the previous subsection and follow the same sequence of update. The results are shared in the next section.

3.5.3. Sanding prevention

Accumulation of coarse solids in the tailings underflow adversely affect the pipe health and can choke the PSV. This phenomenon is known as sanding, and it occurs when the tailings density (ρ_t) increases beyond a certain threshold, causing solids to settle quicker than they can be removed. Through control of the tailings flow rate (F_t), the tailings density ρ_t can be regulated below the sanding threshold, which is the third objective this work looks to optimize. The sanding threshold is given as $1650 kgm^{-3}$ in literature Gilbert (2004). However, in the current study a lower threshold of $1480 kgm^{-3}$ is chosen as a tighter constraint. With a same sampling time of 1 h, the states observed s_t^{SC} are given in Eq. (42). The output of the actor is a probability distribution as shown in Eq. (43) from which the action ΔF_t is sampled every 1 h (Eq. (44)). The action updates the tailings withdrawal flow rate F_t as shown in Eq. (45), where F_{t_s} represents the steady state tailings flow rate.

$$s_t^{SC} = [\rho_t, \rho_{t_{SP}}, F_t(t)] \tag{42}$$

$$\mu_t^{SC}, \sigma_t^{SC} = \pi_{\theta_{SC_A}}(s_t^{SC_A}) \tag{43}$$

$$\Delta F_t(t) \sim N(\mu_t^{SC}, \sigma_t^{SC}) \tag{44}$$

$$F_t(t) = F_{t_s} + \Delta F_t(t) \tag{45}$$

The actual instantaneous rewards are given in equation (46) and further expanded in Eqs. (46) and (47). They would be used to calculate the actual returns used in the critic loss function represented in Eq. (21), which will then be used to calculate the actor loss as shown in Eq. (22).

$$\Delta\rho_t = \rho_t - \rho_{t_{SP}}$$

$$r_t^{SC} = - \mid \Delta\rho_t \mid^2 - \mid \Delta F_t(t) \mid^2 - F_t^{breach} \tag{46}$$

$$F_t^{breach} = \begin{cases} 0, & 0.8F_{t_s} \leq F_t(t) \leq 1.2F_{t_s} \\ \mid F_t - 1.2F_{t_s} \mid, & F_t > 1.2F_{t_s} \\ \mid F_t - 0.8F_{t_s} \mid, & F_t < 0.8F_{t_s} \end{cases} \tag{47}$$

Simulation details and results are provided in Section 4.

3.5.4. Coerced learning

Another novel contribution of this paper is leveraging the existing control strategy to initially teach the RL agent to learn and explore in the stable operational region of the state/action space. This is especially useful when dealing with a nonlinear process such as the PSV. This is an adaptation of the imitation learning concept into this work. The strategy developed has been termed coerced learning and was implemented by learning from an interactive expert demonstrator namely learn from existing control strategy.

The training is carried out in 2 phases. In Phase 1, the action taken by the actor-critic is limited subject to a defined bound of the expert demonstrator's action. This is achieved by adding an additional factor in reward calculation in the first few episodes. In this phase, a demultiplexer chooses between the RL agent's action a_t^{RL} and the expert controller's action a_t^C as given in equation (48) and illustrated in Fig. 6. The RL agent's action is evaluated for the regular reward if it is within $\pm$ 5% of the action that the expert demonstrator would choose for the given measurements. A penalizing mechanism considering the distance of the RL agent's action from the bounds is utilized otherwise. Beyond these bounds, the coerced learning factor *cc* factor penalizes the actions taken by the RL agent. This is represented in equation (49), where *A* represents the complete practical range of actions. The penalty is hence proportional to the deviation between the action taken by the RL agent and the expert demonstrator. If the RL agent's action lies within the acceptable range, the reward is proportional to the deviation from the setpoint in the case of setpoint tracking.

$$a_t = \begin{cases} a_t^C, & 0.95a_t^C < a_t^{RL} < 1.05a_t^C \\ a_t^{RL}, & otherwise \end{cases} \tag{48}$$

$$r_t = \begin{cases} - \mid I_{F-M_{SP}} - I_{F-M} \mid, & 0.95a_t^C < a_t^{RL} < 1.05a_t^C \\ \frac{cc}{A - \mid a_t^C - a_t^{RL} \mid}, & otherwise \end{cases} \tag{49}$$

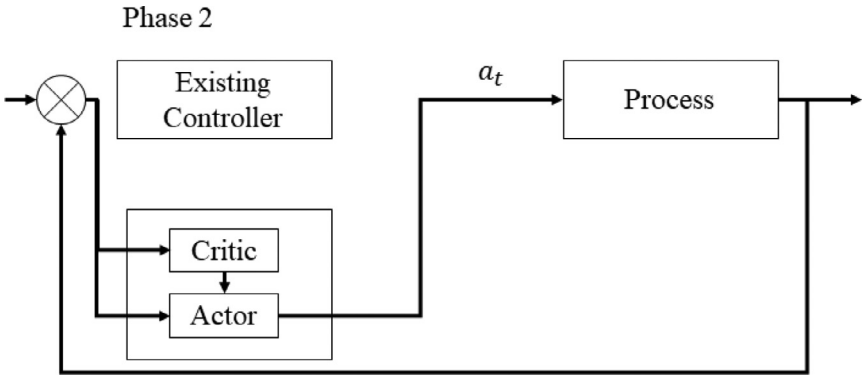

Fig. 7. Phase 2 of Coerced Control Training.

After a specified number of episodes, the training switches to phase 2. In this phase, the RL agent's learning is independent of the expert demonstrator, as shown in Fig. 7. Compared to this, in behavior cloning, the learner assumes the demonstrator's policy to be optimal and aims to imitate it by copying the actions it takes in given states. This is, thus, a semi-supervised learning scheme where the RL agent leverages experience from the expert demonstrator (like a conventional controller) to define the direction for exploration to ensure that the exploration happens within a stable region while control objectives are met. The impact of introducing this factor on the training and on-line execution along with its wider implications for control are discussed in Section 4.

4. Results & discussion

4.1. Infrastructure

For this study, a high fidelity model of the PSV was considered. The PSV simulation as well as the RL code was implemented using Tensorflow v. 1.9.0 in Python 3.7.1. Windows 10 64-bit OS running on a Lambda computer with Intel i9-9820x processor with 20 threads was utilized for the A3C based learning.

Details of the input and output vectors to the actor as well as the critic have been elaborated in Section 3.5 for the hierarchical architecture based agents as well as sanding prevention scheme. Fully connected feedforward neural networks are used as function approximators for both the actor and the critic. There is 1 hidden layer for the actor, in all cases, which contains 200 nodes, and it is fully connected to the output layer, with a nonlinear activation function applied to its output. The output layer of the actor consists of a mean and standard deviation as shown in Fig. 4, from which the actions are sampled. The nodes corresponding to the mean (μ_t) have a *tanh* activation function applied in the output layer. The nodes corresponding to the standard deviation (σ_t) have a *softplus* activation function applied in the output layer. Similarly the critic is structured with an input layer fully connected to 1 hidden layer with 100 nodes using a *tanh* activation function, which is in turn fully connected to the output layer. The sample time for each policy is mentioned in the corresponding sections.

4.2. Low level RL - Interface level control

4.2.1. Setup

To comprehensively illustrate the performance of the RL agent in tracking the froth-middlings interface level setpoint, it is compared to the conventional auxiliary controller taken from Liu et al. (2015). It is a proportional controller with gain $p = -10^{-7}$ given in Eq. (51), where F_{m_s} represents the steady state middlings flow rate. In a similar fashion to the RL agent, its control output takes into account the deviation of the interface level from its setpoint to determine the error term (e_t) as given in Eq. (50).

$$e_t = \Delta I_{F-M}(t) = I_{F-M}(t) - I_{F-M_{SP}}(t) \tag{50}$$

$$F_m = F_{m_s} + p.e_t \tag{51}$$

As mentioned in Section 3.5.4, training occurred over two distinct phases, termed coerced learning. In phase 1 of coerced learning the RL agent's action was limited to a defined bound from the expert demonstrator's (conventional controller) action. In phase 2, it was allowed to explore the action space freely. The sampling time for the control action taken by both the RL agent and the conventional controller is 1 h. The RL agent is trained for a total of 20,000 episodes (constituting 4000 hours each), out of which, the first 1500 episodes are spent in phase 1 and the remaining are spent in phase 2.

4.2.2. Results

Servo tracking based on versatile direction and magnitude changes to interface level setpoint $I_{F-M_{SP}}$ is carried out. The setpoint change is instigated once every 400 hours. All process and manipulated variables are recorded for quantitative assessment.

Without coerced learning, the actions taken by the RL agent in the initial episodes led the PSV to the unstable region from which it could not recover. This hindered learning and eventually the convergence to the optimal policy. Coerced learning enabled the RL agent to learn from the conventional controller to find a stable operating region within an empirically determined number of episodes, as shown in Fig. 8. Subplots (a-d) of Fig. 8 depict a stage wise improvement in the policy gradients towards the conventional strategy. As evident from the subplots, the action selection improves as more training episodes elapse. This corresponds to the RL agent learning to take actions within the stable operating region, denoted by the upper and lower bounds on subplots (a-d) of Fig. 8. Here, simulation 1 had resulted in a policy with action selection within limits well within 500 episodes of learning. Simulation 2 took a long time to converge to satisfy the boundary criteria. Based on this, in order to cater to the worst case scenarios, phase 1 was run for 1500 episodes.

In phase 2, the rewarding structure is based only on the control objective for interface level. Uncertainty of $\pm$ 10% is introduced to the middlings flow rate F_m to correspond to the actuator disturbances in real scenarios. Best policy is based on the best cumulative rewards obtained in any episode in phase 2. As illustrated in Fig. 9, the RL agent has obtained the best projection of the state space into the action space. As evident from the figure and metrics such as mean squared error (MSE) and intergral of absolute error (IAE), the RL agent tracks the setpoint better in comparison to the conventional controller. Furthermore, the ability of the RL

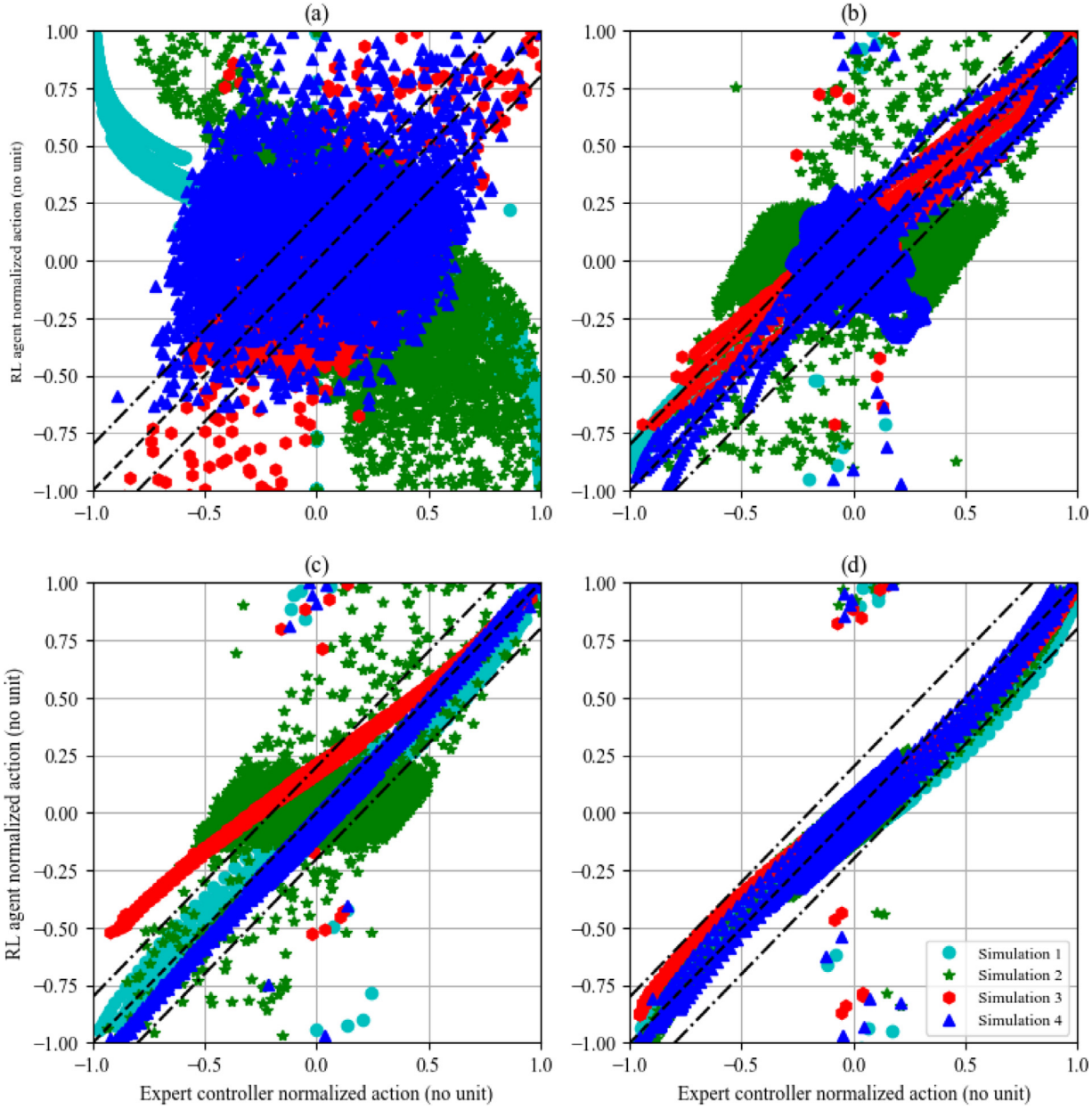

Fig. 8. Phase 1 of Coerced Learning: control action taken by 4 different RL agents relative to expert demonstrator at (a) 0 episodes, (b) 500 episodes, (c) 1000 episodes, and (d) 1500 episodes of training.

agent to track the setpoint in the presence of disturbances relative to the conventional controller is tested. White noise of magnitude $\pm$ 30% of the nominal bitumen content in the ore (α_b^{ore}) is included. The results obtained are displayed in Fig. 10. Hence, the RL agent displays successful servotracking abilities in the presence of varied bitumen content in ore. The RL agent generalizes well over varying operating conditions, controlling the interface level to track the setpoint. The control performance is assessed by MSE, IAE, and variance of control (VC), which are provided in Table 2.

As is also visible in Fig. 9 and Fig. 10, the RL agent significantly outperforms the conventional controller in terms of MSE and IAE as presented in Table 2 The lower MSE conveys the RL agent's ability to maintain lower variance of the interface level I_{F-M} from the setpoint $I_{F-M_{SP}}$ overall while the lower IAE shows that less error is accumulated over time. This is true for both the cases: with constant ore quality and with varying ore quality. This shows the effectiveness of coerced learning in leveraging imitation learning to learn from the conventional controller in the phase 1 of training and eventually outperforming it without the need for model information. The RL agent, however, has a greater variance of control (VC) in both cases. Although the VC is within the acceptable range, this hints that the conventional controller is smoother.

4.3. Supervisory RL - Recovery rate optimization

4.3.1. Setup

The bitumen recovery rate is presented in Section 2.6. The handle used to address the recovery rate *RR* was chosen to be the froth-middlings interface level $I_{F-M_{SP}}$. The sampling interval is 2 hours corresponding to the frequency at which lab samples are available (Section 3.5.2). The action space of the supervisory RL agent is to vary the froth-middlings interface level setpoint $\Delta I_{F-M_{SP}}$ which then prompts the lower level RL agent to manipulate the middling flow rate ΔF_m to track the updated setpoint,

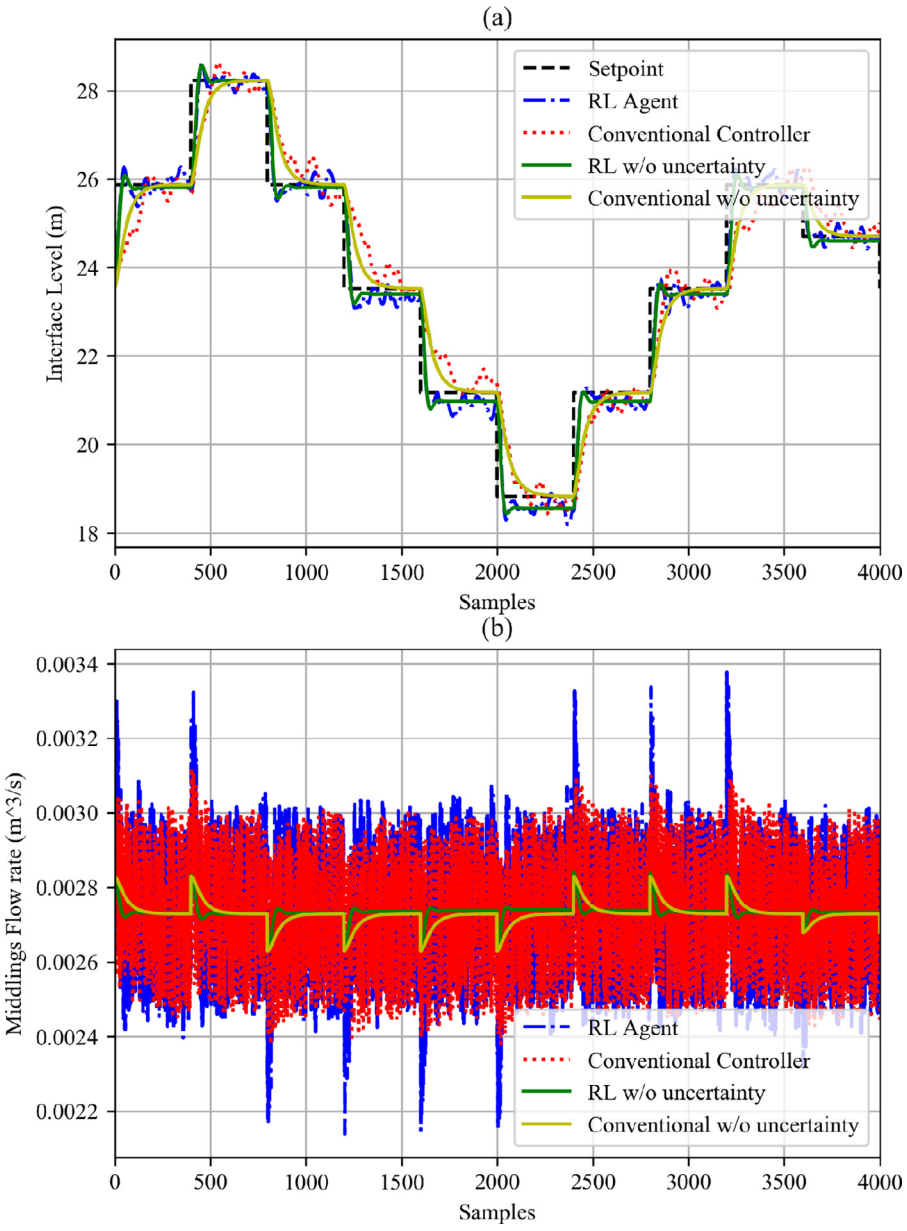

Fig. 9. Froth-middlings interface setpoint tracking results.

completing the hierarchy. A lower level strategy, involving continuous space state/action pair, would require exhaustive exploration. This is evidently achieved by the A3C scheme. However, at the hierarchical level, with a stable lower loop, the agent could possibly require less rigor while training. To understand this, a less data efficient on-policy (policy gradient (DPG)) agent and the off-policy (A3C) agent are deployed here.

4.3.2. Results

Both the RL agents (DPG and A3C) were trained for 10,000 episodes. Here, an episode constitutes of 100 hours with a sampling interval of 1 h. The interface level setpoint I_{F-M} is updated every time a new ore composition data becomes available. The results are displayed in Fig. 11 relative to the regulated interface level. Fig. 11 subplot (a) displays the recovery rate *RR* while Fig. 11 subplot (b) displays the control action $\Delta I_{F-M_{SP}}$ taken by the supervisory RL agents to maximize the recovery rate. Fig. 11 subplot (c) indicates the control action ΔF_m taken by their corresponding low level RL agent to track the updated setpoint. The *RR* peaks above 1 and is explained through the observation that the froth volume (V_f) decreases in accordance with the interface level setpoint changes directed by the supervisory RL agent in the hierarchical control scheme. It then finally settles to a value around the open loop interface level value as the supervisory RL agents in the hierarchical control scheme ordains a final froth-middlings interface setpoint. The overall *RR* relative to the regulated interface level is presented in Table 3.

From Fig. 11 and Table 3, it is clear that the RL based hierarchical control schemes are able to achieve a significantly higher average recovery rate *RR* as compared to the regulated interface level. It is able to do this while maintaining the change in setpoint ($\Delta I_{F-M_{SP}}$), the interface level (I_{F-M}), and the middlings flow rate (F_m) within operational limits. Since there is stable interface tracking at the lower level, the supervisory RL is able to take actions that maintain the PSV in a stable state, leading to a stable

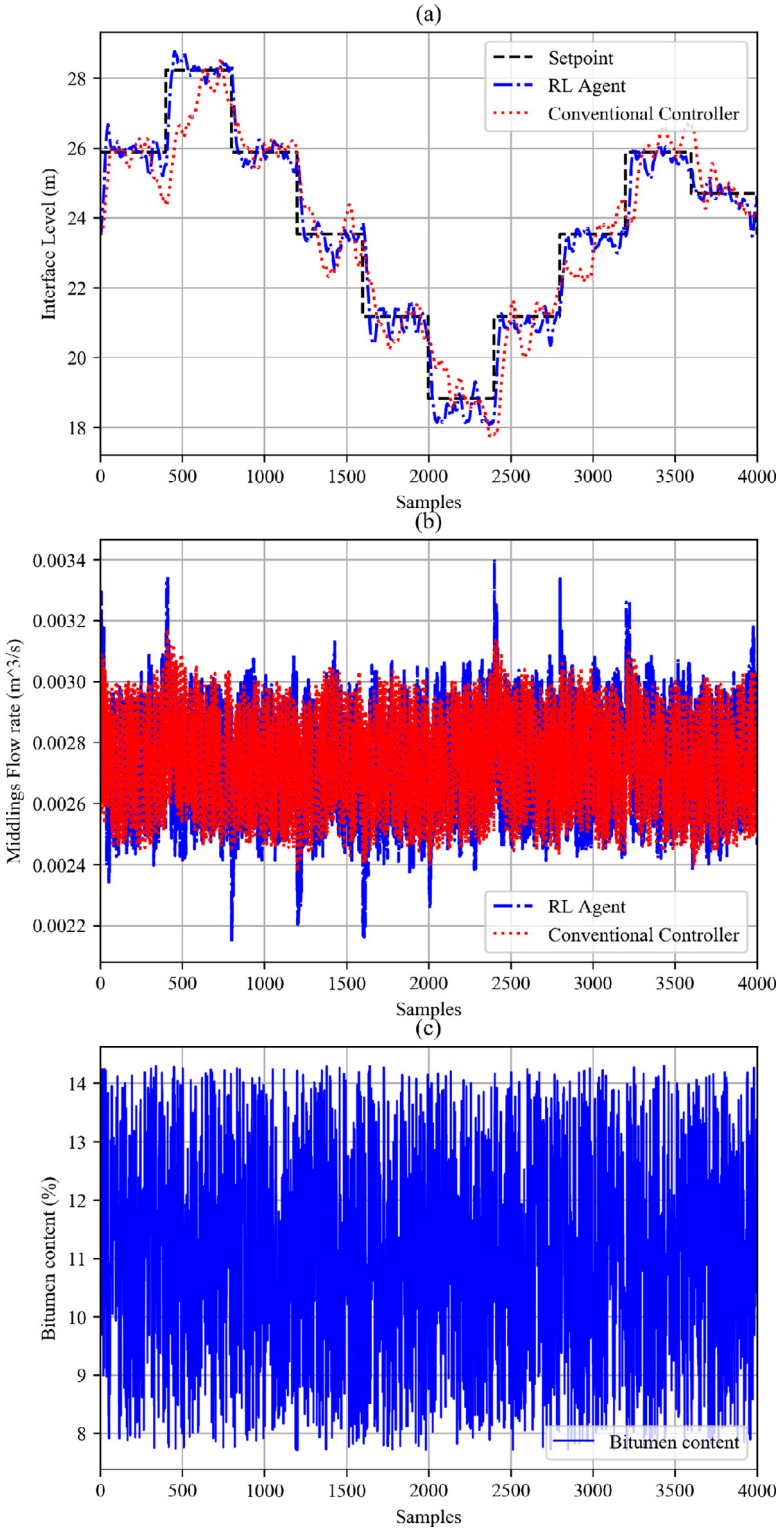

Fig. 10. Froth-middlings interface setpoint tracking results with varying ore quality: (a) Interface level, (b) Middlings withdrawal flow rate, and (c) Ore quality.

Table 2
Froth-middlings interface setpoint tracking with uncertainty results.

	Control Scheme	Mean Squared Error	Integral Absolute Error	Variance of Control
Constant ore quality	RL Controller	0.24	1100.32	3.10e-08
	Conventional Controller	0.64	2154.20	2.54e-08
Varying ore quality	RL Controller	0.31	1408.59	3.23e-08
	Conventional Controller	0.84	2594.64	2.63e-08

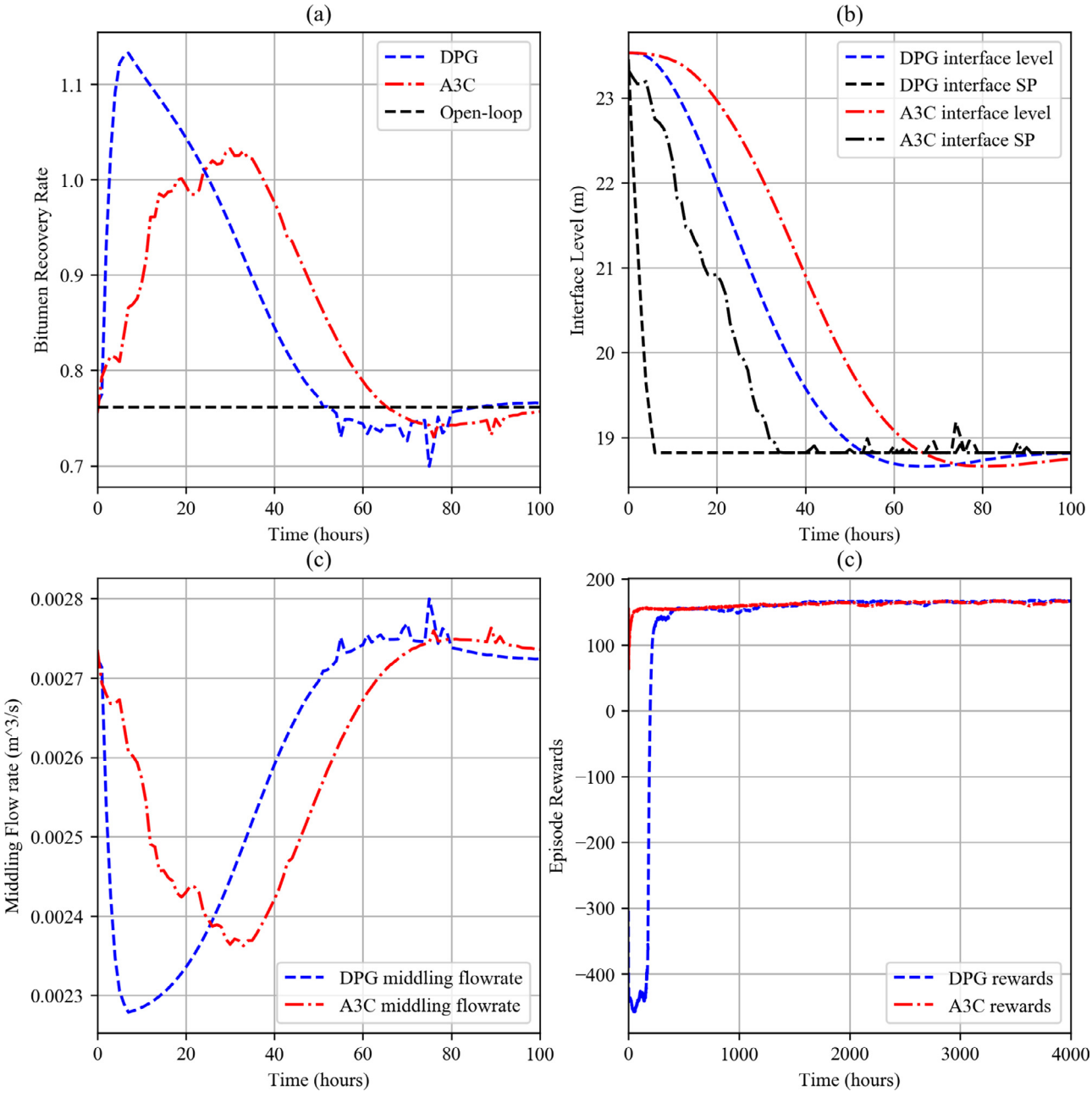

Fig. 11. Recovery rate optimization results: (a) Recovery rate, (b) Interface level, and (c) Middlings withdrawal flow rate, and (d) Training rewards.

Table 3
Recovery rate optimization results.

Control Scheme	Average Recovery Rate
DPG RL Controller	0.8650
A3C RL Controller	0.8653
Open loop	0.76113

RR optimization scheme. Owing to this, both the DPG and A3C agents converged to a similar policy with near optimal performance. However, the DPG agent required more episodes initially to converge. Another interesting observation was that the variance in A3C agent's rewards was 15% higher than that of the DPG agent, indicating active exploration, which is a preferred attribute in RL. This is evident from subplot (b) of Fig. 11. Also, it is possible to infer that the A3C scheme is more sample efficient inherently as compared to the DPG scheme.

4.4. Sanding prevention

4.4.1. Setup

Sanding prevention is implemented as a safety measure to ensure regular PSV function during interface level tracking. The tailings density (ρ_t) is regulated through the tailings flow rate (F_t). This control is activated when the tailings density exceeds a set sanding threshold. The sanding prevention RL agent then manipulates the tailings density by action (ΔF_t) to bring the tailings density below the set threshold. The threshold used in this experiment is set at $1480 kgm^{-3}$. This low level sanding prevention RL agent is built to co-exist with the low level interface level control RL agent reported in Section 4.2. A similar sample time is followed here. The two loops are sequentially executed during simulation.

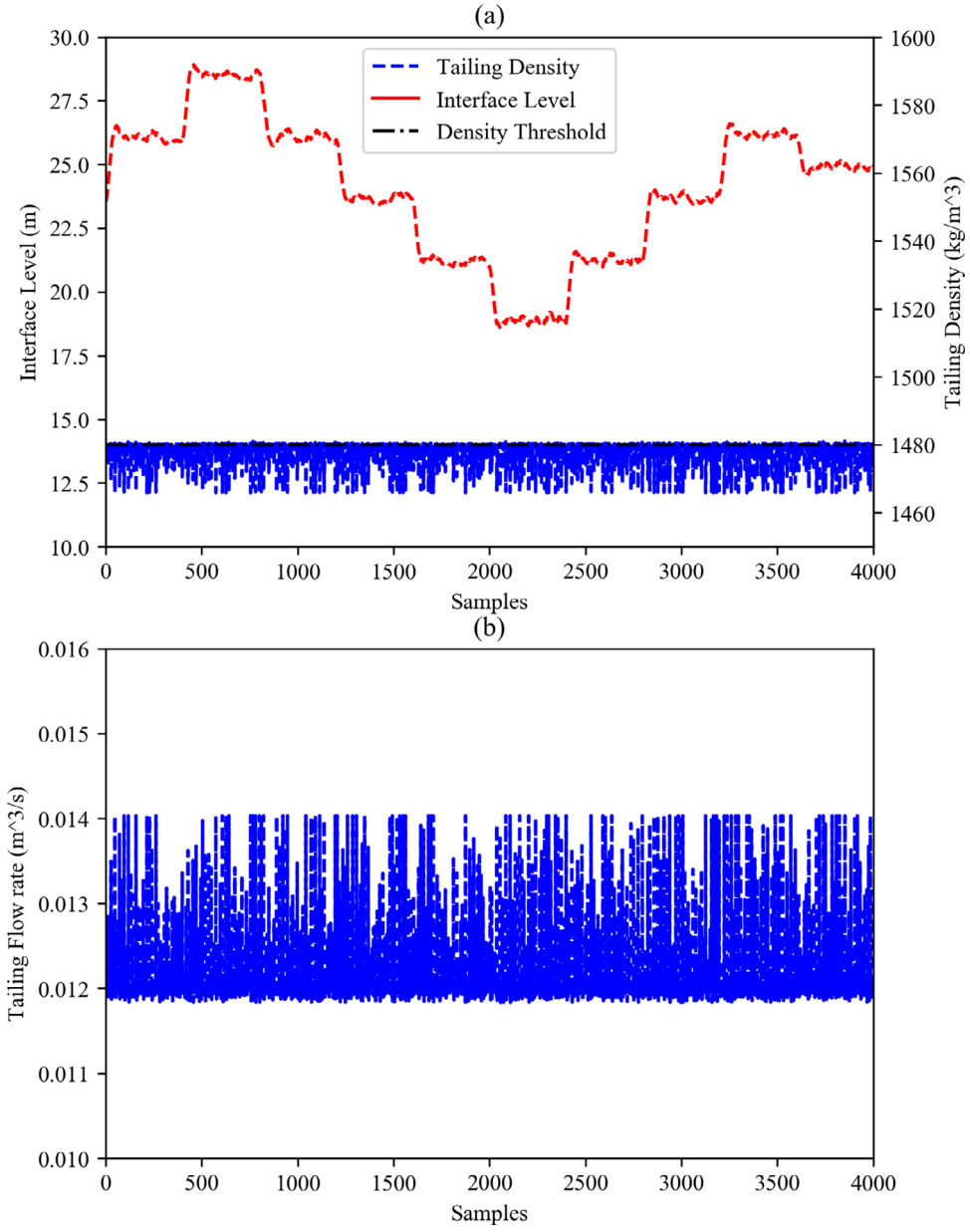

Fig. 12. Sanding prevention results: (a) Interface level and tailings density, and (b) Tailings flow rate.

4.4.2. Results

The sanding prevention RL agent was trained for 20,000 episodes of 4000 hours each. A control action (ΔF_t) was chosen every 1 h. While the setpoint was tracked by the low level interface level control RL agent detailed in Section 4.2 by determining a control action (ΔF_m), the tailing density is controlled through the non interacting low level sanding prevention RL agent determining a control action (ΔF_t) concurrently. The results obtained are shown in Fig. 12. As the subplot (a) of Fig. 12 shows, the low level sanding prevention RL agent is able to successfully bring the tailings density (ρ_t) below the sanding threshold every time it goes beyond the threshold during interface level changes. These correspond to the changes in tailings flow rate (F_t) at the times when the control is activated as shown in subplot (b) of Fig. 12.

5. Conclusions

In this work, a RL based control strategy was developed to implement effective hierarchical control for PSV in presence of disturbances in the middlings flow rate and uncertainty in ore composition. The Supervisory A3C based RL agent manipulated the interface level set point to improve the bitumen recovery rate. The resulting lower level RL agent for servo tracking and ore quality variance oriented regulation of interface level was implemented using an A3C based middlings flow rate manipulation. A sanding prevention scheme was also implemented using a separate A3C based RL agent. The A3C based global RL agents learn the optimal middlings and tailings flow rates to obtain each defined objective. The RL agents map the state space on to the action space through the experience gained by repeated interactions with a high fidelity model of the PSV. The existing conventional control strategy was leveraged using a variant of behaviour cloning, termed as coerced learning. This initially assisted the RL agent in discovering the stable operating region of the action space given the nonlinear nature of the gravity-based separation process. From there, the RL agent was able to independently learn the optimal actions to be taken in the range to achieve its goals. The lower level RL agent for interface level control demonstrated better performance

in terms of IAE and MSE for stable and varying ore quality relative to the conventional controller. The supervisory RL agent also demonstrated a higher bitumen recovery rate than reported with conventional control in keeping with the economic optimization objectives. Furthermore, to evaluate the impact of the hierarchical structure, two different supervisory RL agents, a DPG and a A3C agent were trained. While both agents converged on a near optimal policy due to the stable lower level interface tracking, the A3C based agent displayed faster convergence and higher exploration. The low level RL agent for sanding prevention was also able to maintain the tailings density below the set threshold to prevent sanding amidst tracking the setpoint changes in the interface level.

Declaration of Competing Interest

The authors declare no conflict of interest.

CRediT authorship contribution statement

Hareem Shafi: Conceptualization, Data curation, Formal analysis, Investigation, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Kirubakaran Velswamy:** Conceptualization, Methodology, Formal analysis, Investigation, Validation, Writing - original draft, Writing - review & editing. **Fadi Ibrahim:** Methodology, Writing - review & editing. **Biao Huang:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Resources, Supervision, Writing - review & editing.

Acknowledgements

This work is supported in part by Natural Sciences Engineering Research Council of Canada (NSERC) Industrial Research Chair in Control of Oil Sands Processes, and the Industrial Research Chair Program.

References

CAPP, 2016. Statistical Handbook for Canada 's Upstream Petroleum Industry. 2016-9999 (January) 233. doi:10.4067/S0718-95162017005000034. http://www.capp.ca/publications-and-statistics/publications/275430

Cleveland, C.J., Morris, C., 2014. Handbook of Energy. Volume II_ Chronologies, Top Ten Lists, and Word Clouds, II. Elsevier doi:10.1016/B978-0-12-397219-4.00009-6. http://www.alternative-energy-news.info/technology/fuel-cells/

Concha, F., Almendra, E.R., 1979. Settling velocities of particulate systems, 2. settling velocities of suspensions of spherical particles. Int. J. Miner. Process. 6 (1), 31–41. doi:10.1016/0301-7516(79)90030-9.

CuiY., Zhu, L., Fujisaki, M., Kanokogi, H., Matsubara, T.,, 2018. Factorial kernel dynamic policy programming for Vinyl Acetate Monomer Plant model control. In: IEEE International Conference on Automation Science and Engineering. IEEE Computer Society, pp. 304–309. doi:10.1109/COASE.2018.8560593.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., Legg, S., 2017. Noisy Networks for Exploration.arXiv:1706.10295

GeY., Li, S., Chang, P., 2018. An approximate dynamic programming method for the optimal control of alkai-Surfactant-Polymer flooding. J. Process Control 64, 15–26. doi:10.1016/j.jprocont.2018.01.010.

Gilbert, W.A., 2004. Dynamic Simulation and Optimal Trajectory Planning for an Oilsand Primary Separation Vessel. University of Alberta Thesis.

Government of Canada, 2018. What are the oil sands? | Natural Resources Canada. https://www.nrcan.gc.ca/energy/oil-sands/18089

Heess, N., Wayne, G., Silver, D., Lillicrap, T., Tassa, Y., Erez, T., 2015. Learning continuous control policies by stochastic value gradients. In: Advances in Neural Information Processing Systems, 2015-Janua, pp. 2944–2952.

Kim, J. W., Park, B. J., Yoo, H., Oh, T. H., Lee, J. H., Lee, J. M., 2020. A model-based deep reinforcement learning method applied to finite-horizon optimal control of nonlinear control-affine system. J. Process Control 87, 166–178. doi:10.1016/j.jprocont.2020.02.003.

LeeJ. M., Lee, J. H., 2005. Approximate dynamic programming-based approaches for input-output data-driven control of nonlinear processes. Automatica 41 (7), 1281–1288. doi:10.1016/j.automatica.2005.02.006.

Li, B., Xu, F., Ren, Z., Espejo, A., 2011. Extended abstract: Primary separation vessel interface control. In: Proceedings of the 2011 International Symposium on Advanced Control of Industrial Processes, ADCONIP 2011, pp. 262–264. https://ieeexplore.ieee.org/abstract/document/5930434

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning.arXiv:1509.02971

Littman, M.L., Dean, T.L., Kaelbling, L.P., 2013. On the complexity of solving markov decision problems. CoRR abs/1302.4971. arXiv:1302.4971.

Liu, S., Zhang, J., Liu, J., 2015. Economic MPC with terminal cost and application to oilsand separation. In: Proceedings of the IFAC-PapersOnLine, 28, pp. 20–25. doi:10.1016/j.ifacol.2015.08.151.

Masliyah, J.H., Cluett, W., Oxenford, J., Tipman, R., 1984. Dynamic simulation of a gravity separation vessel. In: Proceedings of the Soc of Mining Engineers of AIME, pp. 145–151.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous Methods for Deep Reinforcement LearningarXiv:1602.01783

Masliyah, J.H., Kwong, T.K., Seyer, F.A., 1981. Theoretical and experimental studies of a gravity separation vessel. Industrial and Engineering Chemistry Process Design and Development 20 (1), 154–160. doi:10.1021/i200012a024.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533. doi:10.1038/nature14236.

Nguyen, T. T., Nguyen, N. D., Nahavandi, S., 2018. Deep Reinforcement Learning for Multi-Agent Systems: A Review of Challenges, Solutions and ApplicationsarXiv:1812.11794

Nian, R., Liu, J., Huang, B., Mutasa, T., 2019. Fault-tolerant control system: a reinforcement learning approach. SICE 1010–1015.

Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P., 2015. High-Dimensional Continuous Control Using Generalized Advantage EstimationarXiv:1506.02438

Shin, J., Badgwell, T.A., Liu, K.-H., Lee, J.H., 2019. Reinforcement learning overview of recent progress and implications for process control. Comput. Chem. Eng. 127, 282–294.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M., 2014. Deterministic policy gradient algorithms. In: Proceedings of the 31st International Conference on Machine Learning, ICML 2014, 1, pp. 605–619.

Spielberg, S., Gopaluni, R., Loewen, P., 2017. Deep reinforcement learning approaches for process control. In: Proceedings of the 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP). IEEE, pp. 201–206. doi:10.1109/ADCONIP.2017.7983780.

Sutton, R.S., Barto, A.G., 2017. Reinforcement learning: an introduction 2018 complete draft. UCL,Computer Science Department, Reinforcement Learning Lectures doi:10.1109/TNN.1998.712192. arXiv:1011.1669v3.

Sutton, R.S., Barto, A.G., Williams, R.J., 1991. Reinforcement learning is direct adaptive optimal control. In: Proceedings of the American Control Conference, 3, pp. 2143–2146. doi:10.1109/37.126844. https://ieeexplore.ieee.org/document/126844/

Swanson, V., 1967. The development of a formula for the direct determination of free settling velocity of any size particle. Trans AIME 238, 160–166.

Swanson, V., 1975. Modification to swansonss free settling equation. Trans AIME 258, 102–103.

Tai, L., Paolo, G., Liu, M., 2017. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems. Institute of Electrical and Electronics Engineers Inc., pp. 31–36. doi:10.1109/IROS.2017.8202134.

Wallis, G., 1969. One-dimensional two-phase flow. McGraw Hill, New York doi:10.1002/aic.690160603.

Wang, Y., Velswamy, K., Huang, B., 2017. A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. Processes 5 (3). doi:10.3390/pr5030046.

Wang, Y., Velswamy, K., Huang, B., 2018. A novel approach to feedback control with deep reinforcement learning. IFAC-PapersOnLine 51 (18), 31–36. doi:10.1016/j.ifacol.2018.09.241.

Watkins, C., Holloway, R., 2014. Technical note : Q-Learning technical note 8 (May), 279–292. doi:10.1007/BF00992698.