Linear-quadratic level control for flotation through reinforcement learning

Norlund, Frida; Tammia, Rasmus; Hägglund, Tore; Soltesz, Kristian

Published in:
IFAC-PapersOnLine

DOI:
10.1016/j.ifacol.2024.08.435

2024

Document Version:
Peer reviewed version (aka post-print)

Link to publication

Citation for published version (APA):
Norlund, F., Tammia, R., Hägglund, T., & Soltesz, K. (2024). Linear-quadratic level control for flotation through reinforcement learning. In *IFAC-PapersOnLine* (Vol. 58, pp. 799-804) https://doi.org/10.1016/j.ifacol.2024.08.435

Total number of authors:
4

Linear-Quadratic Level Control for Flotation through Reinforcement Learning

F. Norlund [*,**] **R. Tammia** [**] **T. Hägglund** [*] **K. Soltesz** [*]

[*] *Lund University, Dept. Automatic Control, Lund, Sweden (email: {frida.norlund,kristian.soltesz,tore.hagglund}@control.lth.se)*
[**] *Boliden AB, Boliden, Sweden (email: rasmus.tammia@boliden.com)*

Abstract: In the mining industry, flotation is a commonly used process to separate valuable minerals from waste rock in a concentrator. The rougher flotation is the first stage of the process and in Boliden AB's concentrator at Aitik, it consists of two lines of four flotation cells each. In this paper we consider one line and the buffer tank upstream of it. Modeling this process step, and maintaining an updated model over time, is a challenge. The process itself changes over time as equipment degrades and parts are replaced. Additionally, the operating conditions in the flotation process change as the ore quality varies. We address these challenges by using reinforcement learning (RL) to design a state feedback controller for level control, without the need of an explicit process model. Using simulations, we compare the performance of the resulting controller to that of the cascade coupled PI-control structure that operates the real plant today. The RL-based controller improves the performance and shows good potential. However, convergence to an admissible control law requires careful hyper-parameter tuning. Industrial deployment thus requires further work to ensure the required reliability.

Keywords: Machine learning methods and applications, Advanced process control, Flotation, Level control, Reinforcement learning.

1. INTRODUCTION

To produce metals, the raw ore needs to be processed in order to concentrate the minerals that will later be smelted into metals. In the process of concentrating minerals, flotation is commonly used. In series of flotation cells, differences in surface properties are used to separate the valuable minerals from waste rock. To do so, the ore is milled and mixed with water to form a slurry to which chemical reagents are added. The reagents make the selected minerals water repellent, which allows them to attach to air bubbles generated at the bottom of the flotation cells and form a mineral froth on top of the slurry in the cells. To extract more of the minerals from the slurry, the tailing from one flotation cell is the feed to the next. The froth is collected as it flows over the rim of the flotation cells. As stated in Bergh and Yianatos (2011), this makes good level control one of the foundations to having good overall recovery of the minerals.

Level control in flotation is traditionally governed by PI controllers. However, since the flotation cells are connected in series, the cell levels form a strongly coupled system (Jamsa-Jounela et al., 2003). The control performance of the PI controllers are often sufficient under normal operating conditions, but not when disturbances, such as inflow variations, enter the system. Therefore multivariable controllers and cascade coupling of SISO-loops are sometimes used. For example, LQ-control and a decoupling controller was investigated by Stenlund and Medvedev (2002). Model predictive control (MPC) has been implemented for flotation by Brooks and Munalula (2017) to control the froth velocity. With this objective, the MPC targets the references to the base layer controllers rather than the level control itself.

A challenge with model-based approaches is that they rely on the development and maintenance of good models. Not only does the ore quality vary over the deposit, which results in different operating conditions, but equipment is also degraded over time and occasionally exchanged which require model updates to maintain good control performance.

This paper investigates reinforcement learning for level control as an example of how a state feedback law can be designed without the need of an explicit process model. The considered process section will be the rougher flotation circuit at the concentrator plant in the Aitik mine, run by Boliden AB located in Gällivare, northern Sweden. The performance of the resulting controller is compared in simulation to the existing control structure of the plant today.

2. PROCESS DESCRIPTION AND MODELING

The considered process consists of a buffer tank and four flotation cells. A schematic picture of the process is shown in Figure 1. The slurry from the two upstream milling lines enters the buffer tank.Its volumetric inflow rate is hence

[⋆] This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors affiliated with the Department of Automatic Control are members of the ELLIIT Strategic Research Area at Lund University.

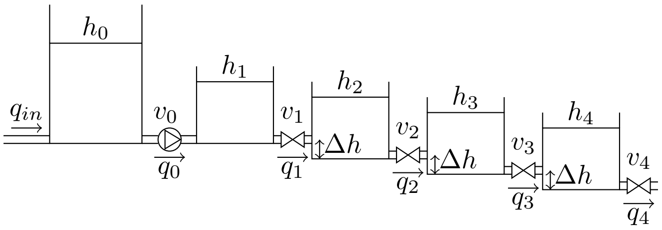

Fig. 1. Schematic overview of the considered process section: a buffer tank followed by four flotation cells in series. Slurry is actively pumped from the buffer tank to the first flotation cell. The slurry level in tank/cell i is h_i, and neighboring flotation cells are mounted at a height difference Δh. The flow q_i out of flotation cell i is moderated by the control signal v_i to a nonlinear valve.

controlled by the milling lines and is therefore considered as a disturbance. The slurry is actively pumped from the buffer tank to the first flotation cell. The flotation cells themselves are of equal size and mounted with Δh height difference between neighbors. Slurry flow between adjacent cells is driven by their difference in hydrostatic pressure, and moderated by a valve, as shown in Figure 1.

For the Aitik plant, it is a fair approximation to assume constant slurry density throughout the process section, and that the combined flow of froth from all cells is negligible compared to the flow of slurry between the cells.

2.1 Buffer tank model

The level in the buffer tank, h_0, will depend on the inflow and outflow of slurry,

$$\frac{dh_0}{dt} = \frac{1}{A_0}(q_{in} - q_0), \tag{1}$$

where A_0, q_{in} and q_0 are the tank cross section area and the tank inflow and outflow rate, respectively. An estimate of the inflow, q_{in}, is available in the control system. The outflow q_0 is related to the control signal, v_0, to the pump that drives the slurry to the first flotation cell. From process data, it can be concluded that q_0 has a linear dependency on v_0. Therefore q_0 is modeled as

$$q_0 = K_f v_0, \tag{2}$$

where K_f is a constant. Combining equations (1) and (2) gives us the model

$$\frac{dh_0}{dt} = \frac{1}{A_0}(q_{in} - K_f v_0). \tag{3}$$

The responses to step changes in the control signal v_0 are shown in Figure 2, both for the resulting model, and the real process.

2.2 Flotation cell model

As for the buffer tank, the level in flotation cell i, h_i, can be modeled as

$$\frac{dh_i}{dt} = \frac{1}{A}(q_{i-1} - q_i). \tag{4}$$

Here A is the cross section area of the flotation cell. For the flotation cells, the outflow, q_i, is approximated by a quadratic function of the control signal v_i. The hydrostatic pressure across the valve will also affect the velocity of the slurry flow between the cells. Since the valve opening is small compared to A, Torricelli's law applies and hence the volumetric flow rate will be proportional to the square root of the slurry level difference between the communicating cells. Thus, q_i can be modeled as

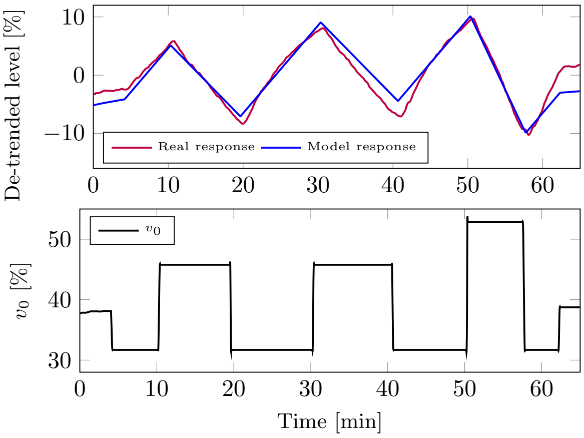

Fig. 2. Step responses of the slurry level resulting from a sequence of steps in the control signal v_0. Red line shows experimental data from the real process; blue line is data from the fitted model (3). The mean value and linear trends are removed from the step responses.

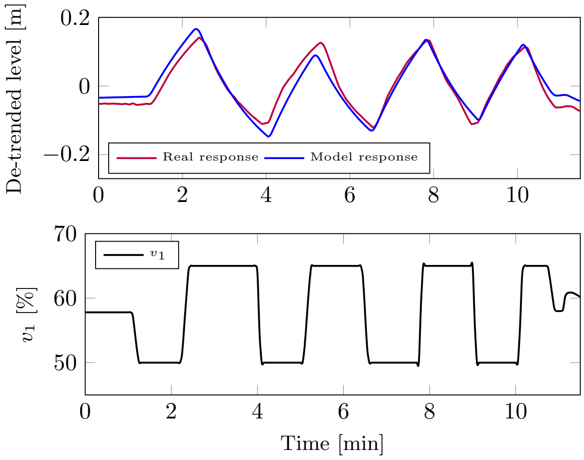

Fig. 3. Step responses of slurry level in flotation cell one, resulting from a sequence of steps in the valve control signal v_1. Red line shows experimental data from the real process; blue line is data from the fitted model (4) with (5), for the first cell. The mean value and linear trends are removed from the step responses.

$$q_i = (c_{2i}v_i^2 + c_{1i}v_i + c_{0i})\sqrt{h_i - (h_{i+1} - \Delta h)}. \tag{5}$$

The constants c_{2k}, c_{1k} and c_{0k} have been chosen to minimize the $\mathcal{L}_2$ error when compared to data from the real plant. A representative example of the resulting model match is shown in Figure 3.

2.3 System model

Combining the models described by equations (3) and (4), we arrive at

$$\frac{dh_0}{dt} = \frac{1}{A_0}(q_{in} - K_f u_0), \tag{6a}$$

$$\frac{dh_i}{dt} = \frac{1}{A}(q_{i-1} - q_i),\ i = 1, \dots, 4, \tag{6b}$$

where $q_0 = K_f u_0$. To represent effects of other phenomena that the model does not capture, a disturbance signal generated by filtering independent white noise sources through auto-regressive (AR) filters was added to the individual levels. The coefficients of the AR filters were chosen to match characteristics of measurements recorded from the real process, for further details on the noise modeling (Norlund, 2022).

3. CONTROL

We investigate the use of a reinforcement learning (RL) algorithm to design a linear quadratic controller. The algorithm is entirely data-driven and has no knowledge of the process model described above. This is of practical interest, since both the process itself, and its operating conditions, change over time. These changes are hard to accommodate online within the outlined model.

The algorithm presented below is based on Bradtke (1992) and Lewis et al. (2012), where the latter also provides a broader introduction to RL. It relies on N samples, collected during closed-loop operation of the system, to update a state feedback controller based on a linear quadratic control cost function. This procedure is iterated until the controller converges to the state feedback law that is optimal for the imposed cost function.

For the traditional linear-quadratic controller, the cost function is given by

$$J(\boldsymbol{x}, \boldsymbol{u}) = \sum_{i=0}^{\infty} \boldsymbol{x}_k Q \boldsymbol{x}_k + \boldsymbol{u}_k R \boldsymbol{u}_k, \tag{7}$$

where $\boldsymbol{u}$ is the sequence of future control signals, and $\boldsymbol{x}$ is the system state. Also note that $\boldsymbol{x}$ and $\boldsymbol{u}$ represent deviations from a reference equilibrium defined by $\boldsymbol{h}_{ref}$ and $\boldsymbol{u}_{ref}$, leading $\boldsymbol{x}$ and $\boldsymbol{u}$ to be controlled to zero and

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{h}_{ref} + \boldsymbol{x}, \\ \boldsymbol{v} &= \boldsymbol{u}_{ref} + \boldsymbol{u}. \end{aligned} \tag{8}$$

The indexing k in (7) refers to time steps and starts from the current sample $\boldsymbol{x}_0$. The matrices Q and R are design matrices chosen to provide an adequate trade-off between state error and control action. The quadratic cost (7) is used to arrive at the linear control law

$$\boldsymbol{u}_k = -L\boldsymbol{x}_k, \tag{9}$$

where the feedback gain L is a constant matrix. The system that is to be controlled can be modeled as

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k, \tag{10}$$

where the matrices A and B represent the dynamics of the system. Traditional linear-quadratic (LQ) control is described further in Glad and Ljung (2000). When the control law is chosen as in equation (9), to control a system represented by (10), the closed loop system will become $\boldsymbol{x}_{i+1} = (A - BL)\boldsymbol{x}_i$. Since this pattern repeats itself for future time steps, we have that

$$\boldsymbol{x}_{k+N} = (A - BL)^N \boldsymbol{x}_k. \tag{11}$$

Using (11), (7) can be written as

$$\begin{aligned} J(\boldsymbol{x}, \boldsymbol{u}) &= \frac{\boldsymbol{x}_k^T}{2} \sum_{i=0}^{\infty} (A^T - L^T B^T)^i (Q + L^T R L)(A - BL)^i \boldsymbol{x}_k \\ &= \frac{1}{2} \boldsymbol{x}_k^T P_L \boldsymbol{x}_k =: \mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k). \end{aligned} \tag{12}$$

Here P_L depends on the chosen feedback gain L. Also note that

$$\mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \frac{1}{2}(\boldsymbol{x}_k^T Q \boldsymbol{x}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k) + \mathcal{Q}(\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1}). \tag{13}$$

Combining equation (10), (12) and (13), $\mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ can be expressed as

$$\begin{aligned} \mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k) &= \frac{1}{2}(\boldsymbol{x}_k^T Q \boldsymbol{x}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k) \\ &\quad + \frac{1}{2}(A\boldsymbol{x}_k + B\boldsymbol{u}_k)^T P_L (A\boldsymbol{x}_k + B\boldsymbol{u}_k) \\ &= \begin{pmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{pmatrix}^T \begin{pmatrix} A^T P_L A + Q & A^T P_L B \\ B^T P_L A & B^T P_L B + R \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{pmatrix}. \end{aligned} \tag{14}$$

Minimizing $\mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ with respect to $\boldsymbol{u}_k$ is done by solving

$$\frac{\partial \mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{u}_k} = \boldsymbol{0}, \tag{15}$$

resulting in

$$\boldsymbol{u}_k = -(B^T P_L B + \mathbf{R})^{-1} B^T P_L A \boldsymbol{x}_k, \tag{16}$$

being a minimizer of $\mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k)$. When a linear model of the system, as in equation (10), is available, $\mathbf{P}_L$ is found by solving an algebraic Riccatti equation. Its solution is then used to calculate the gain matrix of the control law in equation (16).

Now let us move on to the case when the system model is unknown. As further explained in Lewis et al. (2012), equation (14) can be written

$$\mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{pmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{pmatrix}^T \begin{pmatrix} S_{xx} & S_{xu} \\ S_{ux} & S_{uu} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{pmatrix}. \tag{17}$$

Here S_{xx} and S_{uu} are symmetric matrices and $S_{xu}^T = S_{ux}$. Minimizing $\mathcal{Q}$ with respect to $\boldsymbol{u}_k$ results in the control law

$$\boldsymbol{u}_k = -S_{uu}^{-1} S_{ux} \boldsymbol{x}_k. \tag{18}$$

This means that if the matrix elements of S can be estimated from data, then a control law can be determined. Combining equation (13) and (9) and re-arranging gives

$$\mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \mathcal{Q}(\boldsymbol{x}_{k+1}, -L\boldsymbol{x}_{x+1}) = \frac{1}{2}(\boldsymbol{x}_k^T Q \boldsymbol{x}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k). \tag{19}$$

With a data set at hand, the right-hand side of this equation consists of only known variables. By expanding equation (17), one can tell that $\mathcal{Q}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is linearly dependent on the elements of S, and can be written

$$\mathcal{Q}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{u})^T \boldsymbol{\theta} \tag{20}$$

where $\boldsymbol{\theta}$ contains the elements of S and $\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{u})$ contains the corresponding combinations of the elements in $\boldsymbol{x}$ and $\boldsymbol{u}$. For example if the system has one state and one control signal, expanding equation (17) gives

$$\begin{aligned} \mathcal{Q}(x_k, u_k) &= S_{xx} x_k^2 + 2S_{xu} x_k u_k + S_{uu} u_k^2 \\ &= \begin{pmatrix} x_k^2 & 2x_k u_k & u_k^2 \end{pmatrix} \begin{pmatrix} S_{xx} \\ S_{xu} \\ S_{uu} \end{pmatrix} = \boldsymbol{\varphi}(x, u)^T \boldsymbol{\theta}. \end{aligned} \tag{21}$$

Combining (19) and (20) gives

$$\begin{aligned}\left(\boldsymbol{\varphi}(\boldsymbol{x}_k,\boldsymbol{u}_k)-\boldsymbol{\varphi}(\boldsymbol{x}_{k+1},-L\boldsymbol{x}_{k+1})\right)^T\boldsymbol{\theta}&\\ =\frac{1}{2}(\boldsymbol{x}_k^TQ\boldsymbol{x}_k+\boldsymbol{u}_k^TR\boldsymbol{u}_k).&\end{aligned}\tag{22}$$

The $\boldsymbol{\theta}$-vector is the only unknown in (22). To solve for $\boldsymbol{\theta}$, at least as many equations as elements of the $\boldsymbol{\theta}$-vector are required. Equation (22) is valid at every time step, and can hence be formed for each sample to give the required number of equations to solve for $\boldsymbol{\theta}$. Assembling these equations into a system, we arrive at

$$\Phi\boldsymbol{\theta}=Y,\tag{23}$$

where each row of Φ contains the first factor in the left-hand side of (22) for a unique time step, and the rows of Y contain the right-hand side of equation (22) for each time step. From (23), the least squares approximation of $\boldsymbol{\theta}$ can be found according to

$$\hat{\boldsymbol{\theta}}=(\Phi^T\Phi)^{-1}\Phi^TY.\tag{24}$$

With this approximation of the elements in S, the control law can be updated according to equation (18).

4. SIMULATIONS

4.1 Designing a controller with the RL-algorithm

The algorithm derived in Section 3 is summarized in Algorithm 1.

Algorithm 1 Tune a controller with the RL-algorithm

```
$L$: Stabilizing feedback gain
$Q$: State weight matrix for cost function
$R$: Control signal weight matrix for cost function
$\boldsymbol{e}$ : $\boldsymbol{f}(t)$ + Noise (ensuring sufficient exploration)
$N$: Number of samples before updating $L$

while $|L - L_{new}| > tol$ or other stop criteria do
    $L = L_{new}$
    Collect $N$ samples of data with the control law
    $\boldsymbol{u}_k = -L\boldsymbol{x}_k + \boldsymbol{e}_k$
    Construct $\Phi$ and $Y$
    Calculate $\hat{\boldsymbol{\theta}}$ with equation (24)
    Find $L_{new}$ from $\hat{\boldsymbol{\theta}}$ according to equation (18)
end while
```

In the case where the algorithm is applied to a simulation model of the process described in Section 2, the initial, stabilizing feedback gain is chosen to be, $L = I$. For this process, keeping the levels of the flotation cells close to their reference points is the highest priority. Therefore, the weights in Q corresponding to the cell levels have higher values than the weights corresponding to the buffer tank and the weights in R corresponding to the control signals.

To ensure that the samples collected contain enough information about the dynamics of the system, a disturbance component is added to the control signal. The disturbance component, $\boldsymbol{e}_k$, that is designed to perform sufficient exploration on this system, consisted of Gaussian white noise and low frequency square waves. The period of the square wave was chosen to match the speed of the dynamics of the system components. The square waves are only present in one cell at a time, so that the effects of them are seen in the surrounding cells. They are applied from left to right in the process, starting in the buffer tank.

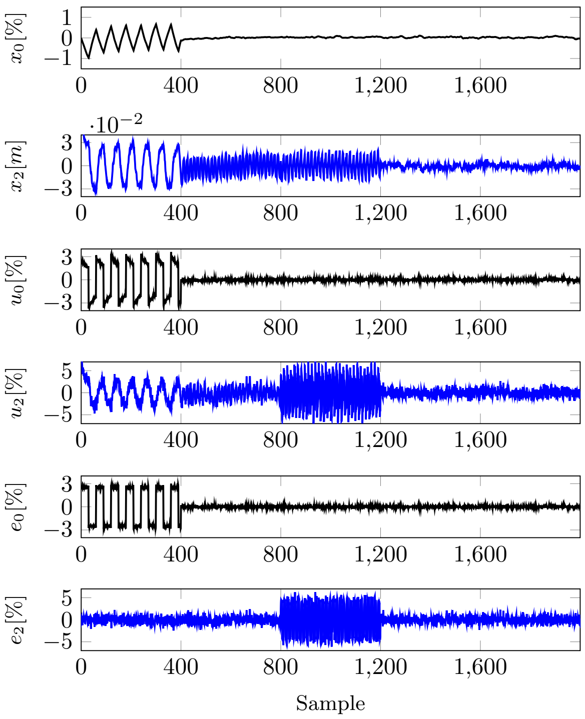

Fig. 4. The first tuning iteration of the RL-algorithm performed on the system consisting of a buffer tank and four flotation cells. There are square waves present in the disturbance term, $\boldsymbol{e}_k$, for one cell at a time, starting with the buffer tank. The buffer tank and the second cell are shown.

Since the system has five states and five control signals, S is a symmetric 10×10 matrix with 55 unknown parameters to be determined. Thus, the minimum number of samples before updating is 56, but to get a better estimate more samples can be gathered. During tuning, $N = 2000$ samples were collected before the control law was updated. The inflow of slurry to the buffer tank was assumed to be constant during the tuning process, as well as the level references for the cells.

In Figure 4, the first tuning iteration is seen for the buffer tank and the second cell. The effects of the disturbances in the buffer tank are clearly seen in the second cell as well. The impact of the square waves in the first cell is also clearly visible in the second cell, they take place between sample 400 and 800.

With five states and five control signals, L will have size 5×5. How the elements of its diagonal are updated over the first 15 iterations of the RL-algorithm is shown in Figure 5. With the control law (9), the diagonal elements of L tells us how much the state x_i contributes to the control signal u_i. Off-diagonal elements in the gain matrix also contribute to the control signals, but theses gains are, in our case, smaller. In the considered, and representative, example, it takes the algorithm seven iterations to converge to a feedback gain.

The controller designed above will drive the states back to the references that were set when it was designed. To enable it to follow other references and correct stationary

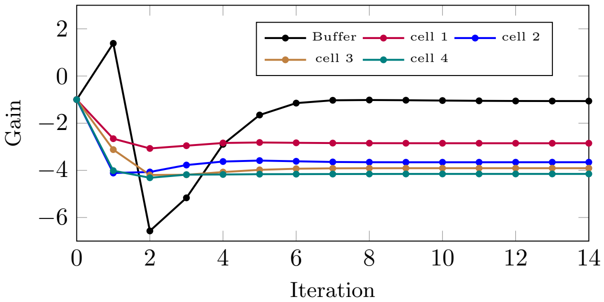

Fig. 5. The RL-algorithm is applied to the system consisting of a buffer tank and four flotation cells. The evolution—iteration by iteration—of the elements on the main diagonal of the gain matrix in the feedback law is visualized as they converge to stationary values.

errors, without re-tuning it, integral action can be added to the controller. This can be achieved by augmenting the state vector with integral states of the tracking error, $\boldsymbol{x}_{int}$, resulting in

$$\boldsymbol{x}_{tot} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_{int} \end{bmatrix}. \tag{25}$$

The feedback gain matrix must also be extended to include feedback gains for the integral states. One way to achieve this is to choose the gain matrix for the integral states, L_{int} to be a diagonal matrix. The total gain matrix will then be

$$L_{tot} = [L, L_{int}]. \tag{26}$$

4.2 RL-based versus currently implemented controller

In the Aitik plant, variations in the slurry flow to the buffer tank is one of the biggest disturbances to the levels in the flotation cells. The biggest inflow disturbance occurs when one of the two milling lines that supply flotation with slurry unexpectedly stops. This roughly halves the inflow of slurry to the flotation series. Process data from a real occurrence of this kind of disturbance has been extracted and used as input to the model. In Figure 6, the RL-based controller's response to the abruptly changed slurry flow is visualized along with the slurry flow. The RL-based controller considered is the resulting controller from the previous section, after the tuning procedure has converged. There is no tuning active when the controller is tested in this section.

In the real plant, the level control is governed by cascade coupled PI-controllers. We have implemented a digital twin of the control system, including actual parameter values, in our simulation environment. This way, the control structure of the real plant and the RL-based controller can be exposed to the same disturbances and their performance can be compared. In Figure 7, the performance of the PI-controllers is shown when the system is exposed to the same inflow disturbance as in Figure 6. While this paper focuses on the comparison between the PI-structure and the RL-based controller, the interested reader can find an evaluation of MPC- and LQ-control for the same process in Norlund (2022).

The level in the buffer tank is not of interest as long as it does not overflow or become empty. It can be seen from Figures 6 and 7 that this requirement is met by both the RL-based controller and the PI-structure.

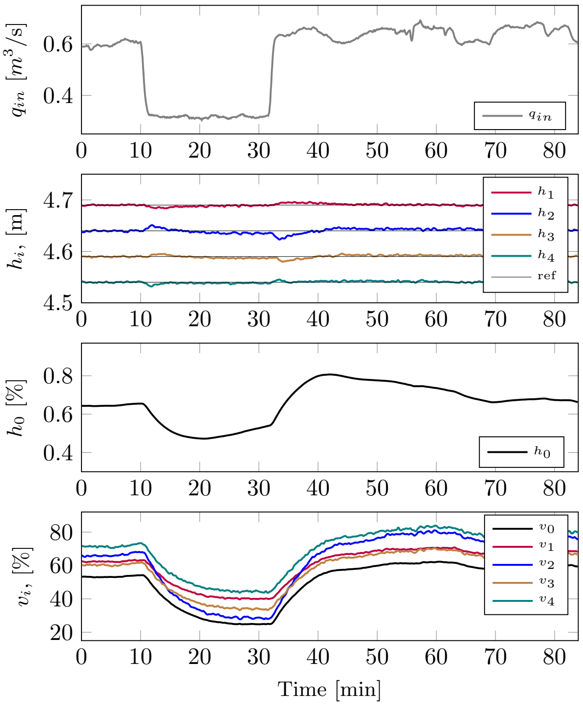

Fig. 6. A slurry inflow, q_{in}, extracted from real process data that contains typical disturbances is fed to the simulation model of the process consisting of a buffer tank and four flotation cells. The system is controlled by the model-free RL-based state feedback controller. The effect of the disturbances for the levels in the flotation cells and the buffer tank are shown along with the corresponding control signals.

The effects on the levels in the flotation cells are of bigger interest. When the big inflow disturbance occurs at time $t = 10$ in Figures 6 and 7, the levels in the cells are affected by it. For the cells, it is desired that the amplitude of the level deviations due to the disturbance should be as small as possible and that the level should return to its reference fast. It is visible in the figures that the impact of the disturbance is bigger for the PI-controllers than for the RL-based controller, both when it comes to the amplitude of the deviations caused by the inflow disturbance, and the duration of the effects of it. Comparing the maximum amplitude deviations in the flotation cells, it is reduced with 42 % on average by the RL-based controller compared to the PI-controllers. When it comes to the time it takes the levels to return to their references, a tolerance around the reference is chosen and the time it takes for the level to return to the tolerated area is measured. This time is on average reduced by 75 % by the RL-based controller compared to the PI-controllers.

Observing the noise canceling properties of the controllers in steady state, it can be observed from Figures 6 and 7 that they both have good noise canceling properties. Looking at the root mean square error (RMSE) for a typical steady state sequence, the RL-based controller reduces it with roughly 50 % compared to the PI-controllers for the flotation cells. However, the noise canceling properties of the PI-controllers is already satisfactory.

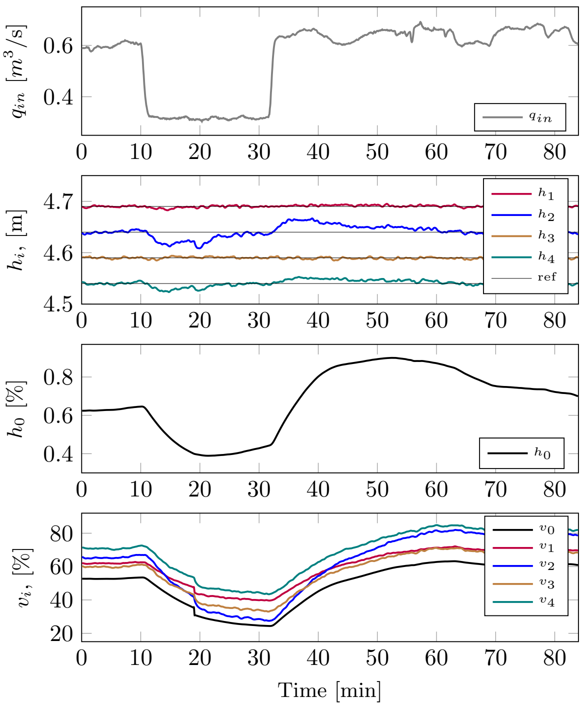

Fig. 7. A slurry inflow, q_{in}, extracted from real process data that contains typical disturbances is fed to the simulation model of the process consisting of a buffer tank and four flotation cells. The system is controlled by PI-controllers configured to be identical to those in the real plant. The effect of the disturbances for the levels in the flotation cells and the buffer tank are shown along with the corresponding control signals.

5. DISCUSSION

As was demonstrated in the previous section, the RL-based feedback controller has overall better performance than the cascade-coupled PI-controllers. It has both better noise canceling properties and better disturbance rejection. This serves to show that the method has potential to improve operation under varying conditions.

Since the algorithm boils down to solving a moderately sized least-squares problem, it can be executed on almost any contemporary computer. However, some considerations, discussed below, need to be made before deployment.

In our studied example, the feedback gain converged over just a few iterations. This requires that the control signal perturbations, $\boldsymbol{e}_k$ provide sufficient exploration.

Even though the algorithm does not require an explicit process model, some knowledge of the process is needed for the choice of the cost function matrices Q and R. For the initial choice of L, and the design of the additive noise, $\boldsymbol{e}_k$ it is also beneficial to have some prior process knowledge.

Since $\boldsymbol{e}_k$ is added to the control signal, the limitations ensuring that $\boldsymbol{u}_k$ stays within its bounds still ensures this for a physical process. If the tuning process is to be performed on the real system, the choice of $\boldsymbol{e}_k$ will also constitute a trade off. It should be chosen as small as possible not to disturb production, while large enough to ensure sufficient exploration. This can lead to that the updated controller performs worse than the previous one, or even destabilises the system.

Looking at the data collection during the tuning procedure from a production point of view, the applied exploratory control signal has a large impact on production. To be practically feasible, further attention would need to be put on how to achieve adequate exploration, while maintaining adequate control performance.

Our study has shown that “model-free” RL-based control can achieve control performance that supersedes that of the currently implemented control system. However, this comes at the cost of initially exciting the dynamics more than would be practically admissible, to obtain sufficient model knowledge. Future work would therefore need to focus on improved experiment designs. Here methods that adjust the experiment online based on the system response, such as Berner and Soltesz (2017) could prove viable.

6. CONCLUSION

Summarizing the above observations, one could conclude that the RL-based control show good potential to design a state feedback controller without the need of an explicit process model. This makes it highly relevant for processes that are hard to model, where the process itself or its operating conditions change over time. However, the approach has a number of practical considerations that must be addressed when applying it to a real process. Particularly, an adequate balance between stable operation during exploration and sufficient excitation of the dynamics must be met.

REFERENCES

Bergh, L. and Yianatos, J. (2011). The long way toward multivariate predictive control of flotation processes. *Journal of Process Control*, 21(2), 226–234.

Berner, J. and Soltesz, K. (2017). Short and robust experiments in relay autotuners. In *22nd IEEE International Conference on Emerging Technologies and Factory Automation*, 1–8.

Bradtke, S. (1992). Reinforcement learning applied to linear quadratic regulation. In *Advances in Neural Information Processing Systems*, 295–302.

Brooks, K. and Munalula, W. (2017). Flotation velocity and grade control using cascaded model predictive controllers. *IFAC-PapersOnLine*, 50(2), 25–30.

Glad, T. and Ljung, L. (2000). *Control theory.* Taylor & Francis, London, UK.

Jamsa-Jounela, S., Dietrich, M., Halmevaara, K., and Tiili, O. (2003). Control of pulp levels in flotation cells. *Control Engineering Practice*, 11(1), 73–81.

Lewis, F., Vrabie, D., and Vamvoudakis, K. (2012). Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, 32(6), 76–105.

Norlund, F. (2022). Comparison of level control strategies for a flotation series in the mining industry. Master thesis report TFRT-6178. Lund University, Department of Automatic Control.

Stenlund, B. and Medvedev, A. (2002). Level control of cascade coupled flotation tanks. *Control Engineering Practice*, 10(4), 443–448.