Journal of Process Control 147 (2025) 103388

Contents lists available at ScienceDirect

Journal of Process Control

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

Bayesian optimization for automatic tuning of a MIMO controller of a flotation bank

Albertus V. Richter [a,b], Johan D. le Roux [a], Ian K. Craig [a,*]

[a] University of Pretoria, Lynnwood, Pretoria, Gauteng, South Africa
[b] Mintek, 200 Malibongwe Dr, Randburg, Gauteng, South Africa

ARTICLE INFO

Keywords:
Automatic tuning
Bayesian optimization
Flotation
Inventory control
Level control
MIMO
PI control
Series tanks

ABSTRACT

A flotation bank consisting of 6 cells in series where each level is controlled by a Proportional–Integral (PI) controller is tuned using Bayesian Optimization (BO) in simulation. A Multi-Input–Multi-Output (MIMO) inventory controller is tuned to optimize the level response of the entire bank. The objective function defining optimality is a trade-off between disturbance rejection and reference tracking in the form of a weighted average of the integral squared error and the integral time absolute error of the level reference tracking error for each cell. The MIMO inventory controller used is a lower diagonal matrix where each element has a PI controller structure. The controller settings selected by the BO are constrained, assuming that the plant is linear, such that only controllers which produce stable closed-loop responses will result. Structured singular value analysis is performed, before tuning, to confirm that this is the case. The BO automated tuner is able to tune multiple PI elements to provide an overall improvement of the flotation bank level control. The method is applied successfully with and without measurement noise on a simulated plant. For use in industry, since the process is simple to model, the controller can be tuned off-line in simulation. To compensate for model-plant mismatch, once the controller is implemented the BO automatic tuner can be allowed a limited number of steps to obtain the optimal controller parameters. This provides a valuable time-saving tool for a process control engineer to tune an industrial plant quickly and efficiently.

1. Introduction

Even though the flotation process has a history that spans a century, long-term supervisory and optimization control of flotation circuits remain elusive [1–4]. As is the case for all hierarchical control structures [5], the success of the upper control layers depend on the lower layers, specifically the accurate tuning of base layer regulatory controllers. The use of Proportional–Integral–Derivative (PID) controllers for regulatory control still dominate the mineral process industry [6]. In light of the frequent disturbances experienced by a mineral processing plant which may alter the dynamics or process requirements, base layer controllers may need to be tuned and re-tuned frequently. The aim of this article is to investigate the use of a data-driven approach to automatically tune the regulatory control layer of an interactive multivariable flotation bank.

The level control of pulp levels in froth flotation banks can range from simple Single-Input–Single-Output (SISO) controllers to more intricate Multiple-Input–Multiple-Output (MIMO) controllers [7]. The main goal of these base layer controllers are to regulate the pulp level of the flotation process [3]. The tighter the base level control of the flotation process, the less interaction in the pulp levels occurs which will benefit the overall grade and recovery. The tuning of these level controllers play an important part in their performance.

The automated tuning of SISO Proportional–Integral (PI) controllers can be done in various ways using e.g. relay tuning [8], reinforcement learning [9], Kalman filtering [10], linear quadratic regulation [11], and Bayesian Optimization [12]. Although SISO PI controllers have low dimensionality, they can become difficult to tune, especially by hand, in a multivariable system with many interactions. Data-driven approaches are receiving increasing attention to address controllers with high dimensionality that are difficult to tune [9,10,12–16].

MIMO systems can be tuned automatically using loop shaping and critical point identification [13,17] or data driven iterative optimization methods [11,14]. One such data driven iterative optimization method is Bayesian Optimization (BO). The goal of BO is to find an optimum according to an objective function while actively optimizing the search direction [18]. BO was successfully implemented in various controller tuning applications [16,19]. More recently it was implemented in mineral processing [12,15,20].

* Corresponding author.
E-mail address: ian.craig@up.ac.za (I.K. Craig).

https://doi.org/10.1016/j.jprocont.2025.103388
Received 21 August 2024; Received in revised form 16 January 2025; Accepted 29 January 2025
Available online 11 February 2025

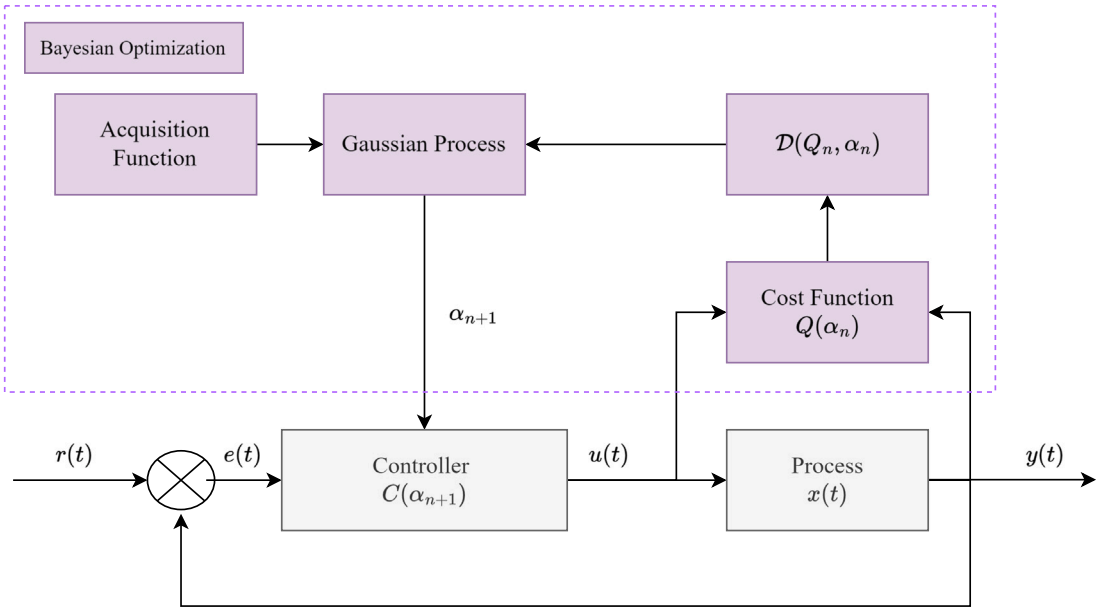

Fig. 1. System diagram of a BO automatic tuner and feedback controller.

Data driven methods are of particular interest for controllers with high dimensionality as they are generally difficult to tune. SISO PI controllers are attractive because of their low dimensionality, but even these PI controllers when combined into a larger MIMO controller can be difficult to tune effectively by hand. The level control of a flotation bank with 6 cells in series is an example of SISO PI loops controlling a multivariable system [7]. There are multiple intricacies in MIMO level control such as controller structure, initializing controller parameters and MIMO closed-loop stability. Apart from ensuring stability and managing the interactions in the system, the challenge is to tune the different loops to ensure satisfactory performance. Robustness can be ensured using μ-analysis [21] (assuming the plant is linear and time-invariant), but this can come at the expense of sufficiently fast controller responses. Placing constraints on the sensitivity transfer function of the closed-loop is shown to be effective in bounding the controller parameters to ensure closed-loop stability while still prioritizing performance [13]. The performance achieved by the MIMO inventory controller shown in [7] with a relatively simple structure of PI controllers is noteworthy. The question remains how to tune this controller automatically.

The contribution of this article is the use of BO to automatically tune a MIMO level controller for a flotation bank consisting of multiple cells in the presence of measurement noise. It extends the work in [22] where only SISO controllers are considered. The aim is to provide a control engineer with a time-saving tool to quickly and efficiently tune a MIMO system.

2. Bayesian optimization and iterative controller tuning

The goal of BO is to iteratively minimize an unknown function, in as few iterations as possible, using a surrogate model of the function to optimize the search direction over the parameter space [23]. In this case the function is the relationship between controller tuning parameters and an objective function which is used to evaluate the performance of the controller within a process. The process has to be perturbed to generate performance data. In BO these perturbations equate to sampling the unknown function that is modelled by the surrogate model. The way the surrogate model is constructed can vary to fit the function being modelled [24]. Here a Gaussian Process (GP) is used for its flexibility and accuracy. The unknown objective function is modelled by the GP, where the variance of the GP is used to model the uncertainty of the GP surrogate model. This is valuable on processes where inherent uncertainty is expected. The BO automated tuner iteratively samples the process and changes the controller parameters to achieve two things:

- minimize the objective function, and
- minimize the number of samples needed.

The response of the controller is influenced by the choice of the objective function. The objective function can be chosen to prioritize different responses such as reference tracking or disturbance rejection.

An acquisition function is used to obtain the next parameters to sample. The acquisition function can be used to prioritize different metrics such as convergence or information gain. With the goal to find the minimum in as few iterations as possible, the Expected Improvement (EI) acquisition function provides a balance between information gain and minimization. The better the surrogate model predicts the actual function, the easier it is to find the parameters that minimize the function. The EI is maximized over the parameter space to obtain the best point to sample. Therefore, the BO algorithm consists of an outer optimization that minimizes the cost function and an inner optimization that maximizes the acquisition function. This is illustrated in Algorithm 1. Fig. 1 shows the interaction between the BO automatic tuner and the controller. The BO automatic tuner requires no structural change to the feedback loop of the controller and only requires access to the performance values and current controller parameters.

Algorithm 1 : Bayesian Optimization

for n = 1,2,...,N **do**
→ select new $\boldsymbol{\alpha}_{n+1}$ by optimizing acquisition function EI:
$\boldsymbol{\alpha}_{n+1} = \arg\max_{\alpha \in \mathcal{A}} EI(\boldsymbol{\alpha}|D)$, (see (8))
→ apply $\boldsymbol{\alpha}_{n+1}$ to the plant and observe the output.
→ query objective function to obtain Q_{k+1} (see (2))
→ augment data $\mathcal{D} = \{\mathcal{D}, (\boldsymbol{\alpha}_{n+1}, Q_{n+1})\}$, (see (3))
→ update statistical model, (see (4) - (7))
end for

2.1. Problem statement

The goal is to find optimal controller settings according to the desired response reflected in the cost function (Q). BO is used to

minimize the cost function in order to select the optimal controller settings. The cost function is set up from the input–output data from the plant together with the current controller settings. The problem is defined as

$$\min_{\boldsymbol{\alpha}\in\mathcal{A}} Q(\boldsymbol{\alpha}), \tag{1}$$

where $\boldsymbol{\alpha}$ is a vector of the parameters from the set $\mathcal{A}$ used to tune the given controller, in this case the proportional and integral parts of the PI controller. The cost function that is minimized by the BO algorithm defines the desired process response. The multivariable controller, expressed as a matrix, is tuned column for column, and the cost function per column can be expressed as

$$Q = \sum_{j}^{M} w_j \beta_j(\boldsymbol{\alpha}), \tag{2}$$

where β_j are the performance variables for $j = 1, \ldots, M$ and ω_j is the weighting of each performance variable. For example, in this study β_j can reflect the influence of the PI gains to the settling time of a flotation cell.

2.2. Gaussian process model

A GP model is used to construct the surrogate model from the input–output data of the process. This model represents the cost function Q. The GP model is constructed using the mean and covariance of the observed data points. The observed data points are collected in a data set $\mathcal{D}$

$$\mathcal{D} = \{\boldsymbol{\alpha}_n, Q_n\}|_{n=1}^{N} \tag{3}$$

where n indicates a pair of data points, starting at 1, and the total number of data points collected through N iterations. The GP model is used to estimate/predict the cost function over the rest of the sample space using a Bayesian linear regression model with zero mean and covariance of the pairs of input–output data with the covariance function $k(\cdot)$ defined as

$$k(\boldsymbol{\alpha}_p, \boldsymbol{\alpha}_q) = \mathrm{cov}(Q(\boldsymbol{\alpha}_p), Q(\boldsymbol{\alpha}_q)), \tag{4}$$

where the subscripts p and q denote different data points in the data set $\mathcal{D}$. The covariance function $k(\cdot)$ used in this work is the standard *ARD Matern 3/2* function. The joint distribution of the prior is given by

$$\begin{bmatrix} \mathbf{Q} \\ \mathbf{Q}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\boldsymbol{A},\boldsymbol{A}) + \sigma^2 \boldsymbol{I} & K(\boldsymbol{A},\boldsymbol{A}_*) \\ K(\boldsymbol{A}_*,\boldsymbol{A}) & K(\boldsymbol{A}_*,\boldsymbol{A}_*) \end{bmatrix}\right), \tag{5}$$

where σ^2 is the variance of the training observations, $\mathbf{Q}$ is the training observations at points $\boldsymbol{A}$ where matrix $\boldsymbol{A}$ consists of all n inputs $\boldsymbol{\alpha}_i$ as column vectors, and $\mathbf{Q}_*$ is the predicted objective function at the test points $\boldsymbol{A}_*$, yet to be trialed, with $K(\boldsymbol{A},\boldsymbol{A})$, $K(\boldsymbol{A},\boldsymbol{A}_*)$, $K(\boldsymbol{A}_*,\boldsymbol{A})$ and $K(\boldsymbol{A}_*,\boldsymbol{A}_*)$ the covariance matrices obtained from the pairs of training data and test points. (In other words, $k(\boldsymbol{\alpha}_p, \boldsymbol{\alpha}_q)$ is the covariance between the two data point vectors $\boldsymbol{\alpha}_p$ and $\boldsymbol{\alpha}_q$, and $K(\boldsymbol{A},\boldsymbol{A}_*)$ is the matrix of covariances between the collection of data points in matrices $\boldsymbol{A}$ and $\boldsymbol{A}_*$.)

The predicted objective function at the test points is

$$\mathbf{Q}_* \mid \boldsymbol{A}, \mathbf{Q}, \boldsymbol{A}_* \sim \mathcal{N}\left(\bar{\mathbf{Q}}_*, \mathrm{cov}\left(\mathbf{Q}_*\right)\right), \tag{6}$$

where

$$\overline{\mathbf{Q}}_* = K(\boldsymbol{A},\boldsymbol{A}_*)^\top \left[K(\boldsymbol{A},\boldsymbol{A}) + \sigma^2 \boldsymbol{I}\right]^{-1} \mathbf{Q} \tag{7a}$$

$$\mathrm{cov}(\mathbf{Q}_*) = K(\boldsymbol{A}*,\boldsymbol{A}_*) - K(\boldsymbol{A},\boldsymbol{A}_*)^\top \left[K(\boldsymbol{A},\boldsymbol{A}) + \sigma^2 \boldsymbol{I}\right]^{-1} K(\boldsymbol{A},\boldsymbol{A}_*). \tag{7b}$$

$\overline{\mathbf{Q}_*}$ is the mean prediction and the variance is the diagonal elements of $\mathrm{cov}(\mathbf{Q}_*)$.

2.3. Acquisition function

The acquisition function provides the next points at which to evaluate the cost function where the acquisition function $\mathcal{L}$ is maximized, with

$$\boldsymbol{\alpha}_{n+1} = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{L}(\boldsymbol{\alpha}|\mathcal{D}). \tag{8}$$

As mentioned earlier, the EI function is used here. The EI function seeks the next evaluation points where the cost function is expected to show the best improvement according to the lowest posterior mean of the GP model. This is given by

$$\mathbf{EI}(\boldsymbol{\alpha}, Q) = \mathbb{E}\max[0, Q_{best}(\boldsymbol{\alpha}_{best}) - Q(\boldsymbol{\alpha})], \tag{9}$$

where $Q_{best}(\boldsymbol{\alpha}_{best})$ is the current best mean at the location $\boldsymbol{\alpha}_{best}$. This is solved analytically as

$$\mathbf{EI}(\boldsymbol{\alpha}) = (Q_{best} - \overline{\mathbf{Q}}_*(\boldsymbol{\alpha}))\Phi(z(\boldsymbol{\alpha})) + \sigma(\boldsymbol{\alpha})\phi(z(\boldsymbol{\alpha})), \tag{10}$$

where Φ is the cumulative distribution function, ϕ is the probability density function and

$$z(\boldsymbol{\alpha}) = \frac{Q_{best} - \overline{\mathbf{Q}}_*(\boldsymbol{\alpha})}{\sigma(\boldsymbol{\alpha})}, \tag{11}$$

where $\sigma(\boldsymbol{\alpha})$ is the predicted standard deviation at α. The derivation of (10) follows from expressing (9) as an integral and applying integration by parts (see Section 3.2 of [20] and [25]).

3. Flotation plant model

3.1. Non-linear plant model

The flotation model used for the flotation bank with 6 cells in series can be found in [7] and [26]. The pulp level in each cell is modelled by integrating the difference between the volume flow in and out of each cell. The plant is given in Fig. 2 under SISO control and in Fig. 3 under MIMO control.

The rate of change in pulp level is given by

$$\dot{h}_i = \frac{\dot{V}_i}{A_i}, \tag{12}$$

where A_i is the cross-sectional area of the cell which is assumed to be constant. Note, (12) assumes a constant gas holdup throughout the bank, similar to [7]. The rate of change in volume for each of the cells $i = 1, \ldots, 6$ is given as

$$\dot{V}_i = B_{i-1} C_v f_c(l_{i-1}) \sqrt{h_{i-1} - h_i + H_{i-1}} - B_i C_v f_c(l_i) \sqrt{h_i - h_{i+1} + H_i}, \tag{13}$$

where C_v is the valve capacity coefficient, l_i is the input signal to the valve, h_i is the pulp level and H_i is the physical height difference between the bottoms of the cells as shown in Fig. 2. H_6 is the difference in height between the bottom of the last cell and a tailings sump. The coefficient B_i is calculated with $l_i = 0.5$ and $Q_F = 2336\ \mathrm{m}^3/\mathrm{h}$ to ensure a steady-state. The rate of change in volume in the first cell is given by

$$\dot{V}_1 = Q_F - B_1 C_v f_c(l_1) \sqrt{h_1 - h_2 + H_1}, \tag{14}$$

and the rate of change in volume in the last cell by

$$\dot{V}_6 = B_5 C_v f_c(l_5) \sqrt{h_5 - h_6 + H_6} - B_6 C_v f_c(l_6) \sqrt{h_6 + H_6}. \tag{15}$$

The valve position $f_c(l_i)$ is assumed to be a linear function of the control input signal adapted around the steady-state position, thus $f_c(l_i) = l_i + 0.5$, with $f_c(l_i) = 1$ being fully open and $f_c(l_i) = 0$ being fully closed. The valve capacity coefficient C_v is given by

$$C_v = 1.17 Q_m \sqrt{\frac{\rho_p}{\Delta p}}, \tag{16}$$

where $Q_m = 1.2\frac{V_{cell}}{\tau/60}$ is the mean volume flow through a cell with a volume of V_{cell}, $\Delta_p = \rho_p g H_i$ is the pressure difference across a valve, ρ_p

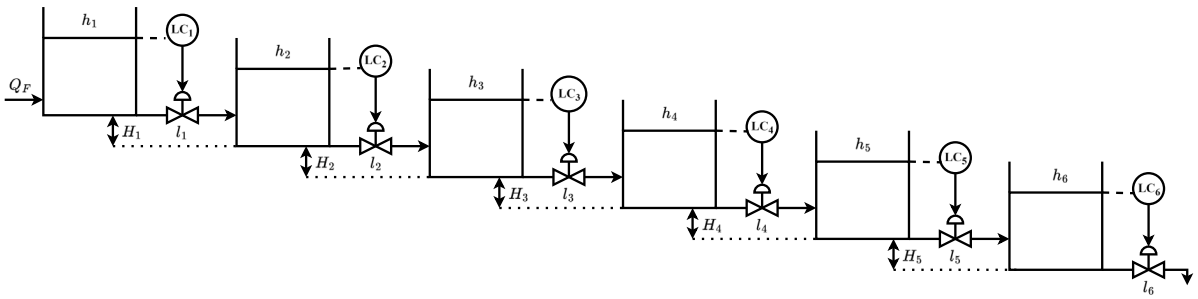

Fig. 2. Flotation bank configuration with individual PI level controllers for each cell.

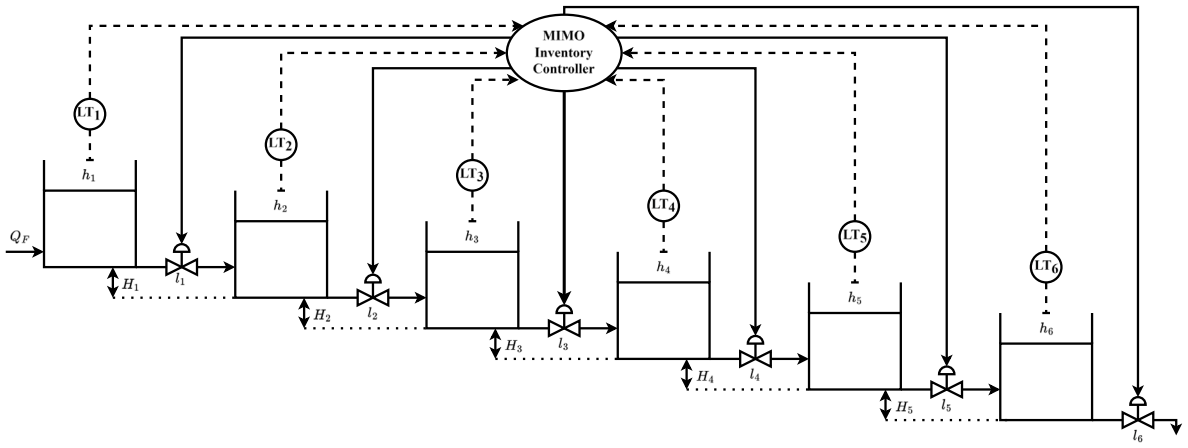

Fig. 3. Flotation bank configuration with MIMO inventory controller.

Table 1
Flotation circuit parameters.

Parameter	Value	Unit	Parameter	Value	Unit
τ	1.50	min	$B_{i=1..5}$	3.49	$m^{2.5}/s$
V_{cell}	76.0	m^3	B_6	1.41	$m^{2.5}/s$
A_i	12.0	m^2	H_i	0.85	m

Table 2
Cell level profile and inputs for steady-state operation.

Variable	Value	Unit	Variable	Value	Unit
h_1	4.06	m	h_4	4.15	m
h_2	4.09	m	h_5	4.18	m
h_3	4.12	m	h_6	4.21	m
Q_F	2336	m^3/h	$l_{i=1..6}$	0.5	–

is the pulp density, and $g = 9.81\ m/s^2$ is the gravitational constant. The pulp retention time in each cell is τ (in minutes). The flotation bank parameters are shown in Table 1 and the initial steady-state operating condition is given in Table 2.

3.2. Linear plant model

The flotation level model is linearized to perform robust stability analysis around a particular operating point, as shown in Section 4.2. The linearization is performed at the operating conditions and written as a state–space model used by the μ-analysis to find the ranges of the controller tuning parameters. Note, the non-linear plant model is controlled in the simulations shown in Section 5 and that the linear plant model is only used for the robust stability analysis in Section 4.2. The non-linear system of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \tag{17a}$$

$$\mathbf{y}(t) = \mathbf{I}\mathbf{x} \tag{17b}$$

is linearized at the equilibrium point shown in Table 2. For (17) the state vector $\mathbf{x}$ represents the six pulp levels h_i, the input vector is $\mathbf{u} = [Q_F, l_{1,\ldots,6}]^T$, the elements of the vector function $\mathbf{f}$ is given by (12), and the output $\mathbf{y}$ is a measurement of the states. The linear representation of (17) is given by

$$\begin{aligned}\dot{\delta}\mathbf{x}(t) &\equiv \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}}\\ \mathbf{u}=\bar{\mathbf{u}}}} \delta\mathbf{x}(t) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}}\\ \mathbf{u}=\bar{\mathbf{u}}}} \delta\mathbf{u}(t)\\ &= \mathbf{A}\delta\mathbf{X}(t) + \mathbf{B}\delta\mathbf{u}(t)\end{aligned} \tag{18a}$$

$$\delta\mathbf{y}(t) = \mathbf{I}\delta\mathbf{x}(t) \tag{18b}$$

where the deviation variables in terms of a steady-state operating condition $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{y}})$ are

$$\delta\boldsymbol{x}(t) = \boldsymbol{x}(t) - \bar{\boldsymbol{x}}$$

$$\delta\boldsymbol{u}(t) = \boldsymbol{u}(t) - \bar{\boldsymbol{u}}$$

$$\delta\boldsymbol{y}(t) = \boldsymbol{y}(t) - \bar{\boldsymbol{y}}$$

and the state–space matrices are given by

$$\mathbf{A} = \begin{bmatrix} -118.7 & 118.7 & 0 & 0 & 0 & 0 \\ 118.7 & -237.4 & 118.7 & 0 & 0 & 0 \\ 0 & 118.7 & -237.4 & 118.7 & 0 & 0 \\ 0 & 0 & 118.7 & -237.4 & 118.7 & 0 \\ 0 & 0 & 0 & 118.7 & -237.4 & 118.7 \\ 0 & 0 & 0 & 0 & 118.7 & -137.9 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0.08333 & -389.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 389.3 & -389.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 389.3 & -389.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 389.3 & -389.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 389.3 & -389.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 389.3 & -389.3 \end{bmatrix}.$$

Note that $\mathbf{B}$ has seven columns for linearization purposes, one for Q_F and six for the valve positions. Q_F is however treated as an uncontrolled input in what follows, hence the dimension of the MIMO inventory controller is 6×6 as shown in Fig. 3.

Table 3
Controller settings for SISO PI obtained using BO (cf. [22]).

Cell	K_c	τ_I (min)	Cell	K_c	τ_I (min)
$i = 1$	−8.08	0.582	$i = 4$	−8.08	0.444
$i = 2$	−8.08	0.409	$i = 5$	−8.08	0.338
$i = 3$	−7.66	0.538	$i = 6$	−6.20	0.565

4. Controller structures, bounds and tuning procedure

4.1. Controller structures

4.1.1. Feedback controller

The SISO controllers to control the plant in Fig. 2 are collected in matrix form as

$$\boldsymbol{C_{PI}} = \text{diag}([c_{11}, c_{22}, \dots, c_{66}]) \tag{19}$$

where each controller has the form

$$c_{ii} = K_{Cii}\left(1 + \frac{1}{\tau_{Iii}}s\right). \tag{20}$$

The controllers are initialized with Skogestad Internal Model Control (SIMC) settings [27], and are tuned further using BO as in [22]. The controller parameters obtained using BO are given in Table 3. Although the cells are uniform with the exception of the last cell, the interactions between the cells depend on their position in the bank. Thus, the optimum parameters found by the BO automatic tuner for the SISO controllers might differ. For example, as seen in Table 3 the gain for cell 3 and 6 differ from the other cells. Intuitively it is reasonable to detune cell 3 to reduce interactions between cells. Since cell 6 is not followed by another cell, control can be allowed to be less aggressive. The SISO controller settings and the linearized plant model of (18) are used to set up an initial parameter space for the μ-analysis in Section 4.2.

4.1.2. Feedback-feedforward controller

It is common for feedforward control to be integrated with feedback control to account for interactions between flotation cells [2,3,7,28]. The inlet flow-rates, if measured, are considered as disturbances and passed through feedforward controllers to reject upstream disturbances in downstream units [29]. However, if the flow-rates are not measured, an inference of the disturbance can be made from the preceding level controller [30].

A feedback-feedforward (FB-FF) controller is defined below where each valve is controlled by its own level plus the upstream level errors, i.e., the error of each cell is used as feedback for its own level and is fed forward to the cells downstream. The purpose of the FB-FF controller is for comparison against the BO automatic tuner settings found for the MIMO inventory controller. The term "feedforward control" is thus used here to indicate that the level errors of upstream cells are used to adjust downstream valve positions.

The FB-FF controller copies the BO SISO PI controllers in (19) through the whole matrix such that the controller is a lower triangular matrix of the form

$$\mathbf{C_{FF}} = \begin{bmatrix} c_{11} & 0 & 0 & 0 & 0 & 0 \\ c_{11} & c_{22} & 0 & 0 & 0 & 0 \\ c_{11} & c_{22} & c_{33} & 0 & 0 & 0 \\ c_{11} & c_{22} & c_{33} & c_{44} & 0 & 0 \\ c_{11} & c_{22} & c_{33} & c_{44} & c_{55} & 0 \\ c_{11} & c_{22} & c_{33} & c_{44} & c_{55} & c_{66} \end{bmatrix}, \tag{21}$$

with controller settings given in Table 3. Since the cells within an industrial flotation bank are generally uniform, it is reasonable to use the controllers on the diagonal to complete the lower triangle of the matrix. This FB-FF structure is similar to the controller used in [7], [26] and [31]. This option is attractive for defining the parameter space used for tuning the MIMO controller as it already gives the final controller structure. The issue of closed-loop stability using this controller is addressed in Section 4.2.

4.1.3. MIMO inventory controller

The inventory controller (cf. [7], [26] and [31]) in Fig. 3 is a MIMO controller that takes into account the total inventory of the flotation bank. Since the entry-ports of the cells are at the bottom, the level of a cell influences both the in- and outflow volume. To ensure local consistency of control loops and following the radiating rule [32,33], a lower triangular matrix structure is chosen for the MIMO inventory controller with the elements of the matrix individual PI controllers as per (20). This choice is the industry standard approach as seen in [7] and [31]. The lower triangular controller is further motivated by the results in [22], which show that sufficiently fast controllers on the diagonal mitigate most of the pushback caused by the lower cells. The reason for this is that the pushback is much slower than the dynamics introduced by the valve. The pushback is governed by the level of the lower cell, which is an integration of the change in flow through the lower cell. However the downstream dynamics is governed by the valve which directly influences the flow into the next cell. Thus, the lower triangular controller structure is required to compensate for the fast acting dynamics of the upstream valves. The upper triangular part of the controller matrix is not required as the fast control on the diagonal sufficiently deals with the interactions that would have been controlled by the upper triangular part of the controller. The controller structure is

$$\mathbf{C_{MIMO}} = \begin{bmatrix} c_{11} & 0 & \cdots & 0 \\ c_{21} & c_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_{61} & c_{62} & \cdots & c_{66} \end{bmatrix}. \tag{22}$$

As shown by the subscripts, the difference between (21) and (22) is that in (21) all elements of a column are equal, whereas in (22) the off-diagonal elements can be different from the diagonal elements. For (21), once the controller parameters for the diagonal elements are defined, the full matrix is defined. For (22), the controller parameters for each element in the matrix must be obtained.

The diagonal elements in (22) control the response of each valve to its own level, whereas the off-diagonal elements influence the interactions between cells. When the diagonal elements are tuned at the same time as the off-diagonal elements, large fluctuations in the level responses occur. The reason for this is that the BO automatic tuner searches over the whole stable parameter space given in Table 4, found using μ-analysis as described in Section 4.2, which include controller parameter combinations that result in under-damped or slow closed-loop responses.

The combination of an under-damped or slow controller on the diagonal and a much faster off-diagonal element means that down-stream cells might e.g. overcompensate for the level fluctuations in cell 1. For example, if the first column in the controller matrix is being tuned, in other words the controllers responsible for providing the control signal to all valves affected by changes in level 1, the control action of valve 1 will influence the flow to all down-stream cells. The off-diagonal elements are supposed to correct for the change in flow induced by valve 1. If both the diagonal and off-diagonal elements change, the plant may encounter large level fluctuation as the controllers may produce widely varying responses in the valve positions. This will slow down the learning part of the BO automatic tuner.

For these reasons a 2-step approach is chosen to first fix, i.e., determine and set, the diagonal controllers, and secondly tune the off-diagonal controllers. By fixing the diagonal controller settings first, the fluctuations are limited, and the tuning procedure also becomes more efficient and less perturbing to normal operation. This can be done by obtaining the controller settings for the diagonal elements using either a classical approach such as SIMC, or a machine learning approach such as BO.

The off-diagonal controller entries in (22) are chosen to be of the form shown in (20) (similar to [20]), with the initial K_{Cij} and τ_{Iij}

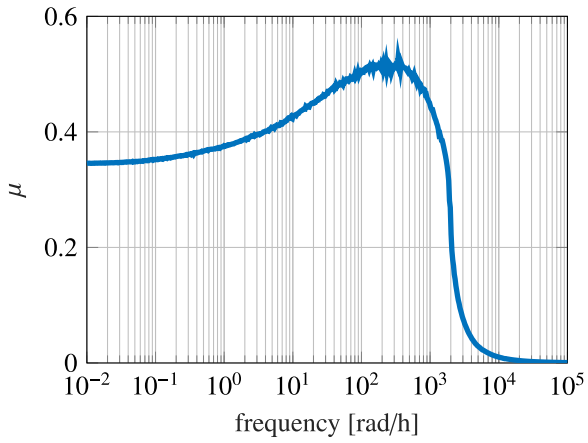

Fig. 4. Maximum singular values (μ) for constraints given in Table 4.

Table 4
MIMO bounds for BO.

Cell (ij)	K_{Cij}		τ_{Iij} (min)	
	Lower	Upper	Lower	Upper
$i1$	−8.79	−1.72	2.022	0.104
$i2$	−8.78	−1.72	1.43	0.0738
$i3$	−8.33	−1.63	1.87	0.0978
$i4$	−8.78	−1.72	1.73	0.0795
$i5$	−8.78	−1.72	1.26	0.0607

values of each column taken from the diagonal controller element c_{ii}, i.e. initially $c_{21} = c_{31} = c_{41} = c_{51} = c_{61} = c_{11}$. This initial structure is equivalent to the FB-FF controller in (21). Initializing the search in this way helps the BO automatic tuner to find the optimal settings in fewer iterations, but does not influence the final settings that are found by the BO.

The outline for the tuning procedure of the MIMO inventory controller in (22), is given as follows:

1. determine the diagonal controller settings and set them constant,
2. find the controller bounds making use of the robust stability analysis and using the FB-FF controller structure as shown in (21), and
3. tune the off-diagonal controller settings using BO.

4.2. Controller bounds

The MIMO inventory controller in (22) is subjected to a robust stability analysis (see e.g. [21]) from which stability bounds on the controller gains (K_{Cij}) and integral time constants (τ_{ij}) are obtained. (The *robstab* function in MATLAB is used to obtain the controller bounds.) For the robust stability analysis the plant is represented by the linear plant model in (18). The controller has a lower diagonal structure with each nonzero element as given in (20). The robust stability analysis is performed similar to [20], with the resulting controller parameter ranges given in Table 4. The result of the robust stability analysis can be visualized through a structured singular value (μ) plot of the closed-loop system shown in Fig. 4. The closed-loop system is robustly stable ($\mu < 1$) over the whole frequency range, meaning that the MIMO controller in (22) will result in a stable closed-loop response for any controller parameter in the ranges shown in Table 4. Constraining the search space of the BO in this way will ensure closed-loop stability as long as the plant is adequately represented by (18).

5. BO controller tuning

The plant and controller is simulated in MATLAB and Simulink. Note, the non-linear plant model in (17) is used to represent the plant during simulations. The diagonal controllers are fixed using the BO settings in Table 3 (cf. [22]). The off-diagonal elements are tuned one cell at a time, using step changes in the level reference values and evaluating the error between the reference and process variable. The controllers are tuned column by column because the plant is perturbed by level setpoint step tests. Only one reference is stepped at a time to clearly differentiate how much interaction is caused by which cell. The reference errors for each cell can be evaluated according to the integral squared error (ISE), the integral absolute error (IAE), or the integral time absolute error (ITAE) [34]. These metrics are defined as

$$ISE(i) = \int_0^T e_i^2(t)dt \tag{23a}$$

$$IAE(i) = \int_0^T \left|e_i(t)\right| dt \tag{23b}$$

$$ITAE(i) = \int_0^T t\left|e_i(t)\right| dt \tag{23c}$$

where i refers to the specific cell evaluated and T is the time period across which the error signal is evaluated.

The errors are collected in an objective function Q, shown below, expressed as a weighted sum of the ISE and ITAE for each cell i. The ISE metric was selected to penalize large errors, whereas the ITAE metric was selected to increasingly penalize errors as the end of the evaluation period $[0, T]$ is approached. The weights of the performance metrics were obtained in a trade-off between these two metrics.

$$Q = \sum_{i=1}^{6} 0.2\text{ISE}(i) + 0.8\text{ITAE}(i). \tag{24}$$

The BO tuner is given 80 iterations to minimize the objective function in terms of the controller parameters. The standard *bayesopt* function in MATLAB is used with the "Expected-improvement" (EI) chosen as the acquisition function and with an exploration ratio of 0.8. The exploration ratio is a parameter used to control the algorithm's propensity to explore the parameter space if little improvement is being made in subsequent iterations. The rest of the *bayesopt* parameters are kept to their default values.

The tuning starts at cell 1 and ends at cell 5 as only the off-diagonal elements are tuned. The tuning is done using two different strategies:

- **Strategy 1**: The off-diagonal elements are added as each next cell is tuned.
- **Strategy 2**: The tuning starts with the full controller of (22).

For Strategy 2 the assumption is that by tuning with all controller elements in place the cost function will be more representative of the final response and the need for doing multiple passes through the columns will be mitigated. For example if the first column is tuned without the full down-stream controllers in place the ability of the down-stream cells to handle the inventory being sent down will be very limited and in turn limit the upstream controllers to more conservative settings. To implement Strategy 2, it is assumed that all cells have similar dynamics and the off-diagonal controllers can be initialized safely with the diagonal controller parameters. This is not necessarily the case for an industrial plant where cells dynamics may differ. In comparison, Strategy 1 is a cautious approach where controllers are added as the automatic tuner progresses down the flotation bank. The computational complexity and optimization time remain the same for both strategies as the number of parameters optimized at a time are the same. Only the cost function changes between the two strategies.

The FB-FF controller serves as the base case for comparison and the BO automatic tuner will aim to find a controller similar to the FB-FF controller. The performance of the BO tuned MIMO controllers are compared to the FB-FF controller using the following performance metrics: IAE, ISE and ITAE.

5.1. Tuning without measurement noise

The initial controller structure is similar to (22)

$$\mathbf{C_{BO}} = \begin{bmatrix} c_{11} & 0 & 0 & 0 & 0 & 0 \\ c_{21} & c_{22} & 0 & 0 & 0 & 0 \\ c_{31} & c_{32} & c_{33} & 0 & 0 & 0 \\ c_{41} & c_{42} & c_{43} & c_{44} & 0 & 0 \\ c_{51} & c_{52} & c_{53} & c_{54} & c_{55} & 0 \\ c_{61} & c_{62} & c_{63} & c_{64} & c_{65} & c_{66} \end{bmatrix} \tag{25}$$

with each element an individual PI controller as given in (20). The diagonal elements in blue are fixed and not tuned. The controller settings found by the BO tuner for Strategies 1 and 2 are shown below in terms of the PI controller gains and integral time constants.

- **Strategy 1**:

$$\mathbf{K}_{ij}^{(1)} = \begin{Bmatrix} -8.08 & 0 & 0 & 0 & 0 & 0 \\ -8.45 & -8.08 & 0 & 0 & 0 & 0 \\ -8.23 & -7.36 & -7.66 & 0 & 0 & 0 \\ -8.61 & -7.52 & -6.53 & -8.08 & 0 & 0 \\ -8.85 & -7.73 & -7.05 & -7.43 & -8.08 & 0 \\ -8.29 & -7.86 & -6.85 & -7.47 & -7.83 & -6.20 \end{Bmatrix} \tag{26a}$$

$$\boldsymbol{\tau}_{Iij}^{(1)} = \begin{Bmatrix} 0.579 & 0 & 0 & 0 & 0 & 0 \\ 1.45 & 0.409 & 0 & 0 & 0 & 0 \\ 1.911 & 0.678 & 0.538 & 0 & 0 & 0 \\ 1.293 & 0.653 & 0.836 & 0.444 & 0 & 0 \\ 1.035 & 0.707 & 0.778 & 0.697 & 0.338 & 0 \\ 1.941 & 0.665 & 0.837 & 0.720 & 0.596 & 0.565 \end{Bmatrix}. \tag{26b}$$

- **Strategy 2**:

$$\mathbf{K}_{ij}^{(2)} = \begin{Bmatrix} -8.08 & 0 & 0 & 0 & 0 & 0 \\ -8.08 & -8.08 & 0 & 0 & 0 & 0 \\ -8.08 & -7.84 & -7.66 & 0 & 0 & 0 \\ -8.08 & -7.61 & -7.42 & -8.08 & 0 & 0 \\ -8.08 & -8.40 & -7.77 & -7.10 & -8.08 & 0 \\ -8.08 & -8.06 & -7.75 & -7.34 & -7.81 & -6.20 \end{Bmatrix} \tag{27a}$$

$$\boldsymbol{\tau}_{Iij}^{(2)} = \begin{Bmatrix} 0.579 & 0 & 0 & 0 & 0 & 0 \\ 0.579 & 0.409 & 0 & 0 & 0 & 0 \\ 0.579 & 0.704 & 0.538 & 0 & 0 & 0 \\ 0.579 & 0.706 & 0.731 & 0.444 & 0 & 0 \\ 0.579 & 0.625 & 0.738 & 0.754 & 0.338 & 0 \\ 0.579 & 0.608 & 0.730 & 0.755 & 0.594 & 0.565 \end{Bmatrix}. \tag{27b}$$

The plant and MIMO controllers are subjected to two different performance tests:

- **Test 1**: a sequence of reference steps of 3 cm to the level of each cell.
- **Test 2**: a −20% disturbance to the feed flow, Q_F at time t = 10 s, as well as 75 m^3/h of spillage water pumped into cell 3 from time t = 100 s.

The performance of the BO controllers are compared to the FB-FF controller. The level responses to the step changes in the reference signals are shown in Fig. 5. Although the controller responses are visually indistinguishable, the performance metrics in Table 5 show that both the BO tuned MIMO controllers perform better than the FB-FF controller for reference tracking. The BO controllers eliminate the error quicker than the FB-FF controller.

The level responses to the disturbances are shown in Fig. 6. Similar to the case above, the controller responses are visually indistinguishable, Table 6 shows that the BO controller tuned according to Strategy 2 performs better than the FB-FF controller in terms of disturbance rejection. However, Strategy 1 performs much worse than both the FB-FF controller and the settings found by Strategy 2. This makes sense as Strategy 1 tunes the controller without all the elements in place, and might be optimal at the time it is being tuned but at the end is not optimal anymore. Once again the smallest improvement is seen in the ISE, similar to what is seen for reference tracking in Table 5.

Table 5
Reference tracking performance.

Metric	Controller			Improvement	
	FB-FF	BO1	BO2	BO1	BO2
IAE	65.4	57.3	57.7	12.39%	11.77%
ISE	34.2	33.4	33.5	2.34%	2.05%
ITAE	17 645	16 120	16 274	8.64%	7.77%
Q	14 123	12 907	13 031	8.61%	7.73%

Table 6
Disturbance rejection performance.

Metric	Controller			Improvement	
	FB-FF	BO1	BO2	BO1	BO2
IAE	60.3	101.4	59.3	−68.2%	1.66%
ISE	57.1	61.0	56.9	−6.83%	0.35%
ITAE	3759	6057	3603	−61.1%	4.15%
Q	3019	4857	2894	−60.88%	4.14%

The results show that the BO automatic tuner is able to tune multiple PI elements to a satisfactory level. The most parameters tuned at once is for cell 1 where 5 PI controllers are tuned (c_{11} is fixed) consisting of 2 parameters each, thus 10 parameters in total. The objective function in (24) is shown to be effective as the minimization of the objective function leads to an improvement over the FB-FF controller for the entire bank in the case of Strategy 2.

The improvements in Tables 5 and 6 are quite small. Because the cells in the simulation have similar dynamics throughout the flotation bank, it is difficult to improve upon the FB-FF controller. However, if the cell dynamics differ and change, as can be expected on an industrial plant, the FB-FF controller settings are not necessarily the best to ensure good performance, especially in terms of interactions between cells. The BO automatic tuner will be able to compensate for these differences and find the best controller settings according to the cost function.

5.2. Tuning with measurement noise

The process above was repeated using Strategy 2 with noise added to the cell level measurements. The magnitude of the reference steps were kept at 3.0 cm. Band limited white noise was selected to simulate level sensor noise of up to ±10 mm in magnitude. This equates to 11% of the operating range of the levels, which is in line with the average resolution of common radar levels sensors such as the VegaPuls 31 [35]. Therefore, the noise is similar to what is expected on an industrial plant. The automatic tuning procedure was conducted with the noise present providing new controller parameters. The final setpoint following and disturbance rejection performance is shown in Figs. 7 and 8 for the controller parameters in (28).

$$\mathbf{K}_{ij}^{(noise)} = \begin{Bmatrix} -8.08 & 0 & 0 & 0 & 0 & 0 \\ -7.53 & -8.08 & 0 & 0 & 0 & 0 \\ -3.02 & -4.55 & -7.66 & 0 & 0 & 0 \\ -5.16 & -3.94 & -5.00 & -8.08 & 0 & 0 \\ -7.68 & -6.32 & -6.00 & -8.30 & -8.08 & 0 \\ -6.22 & -6.88 & -8.48 & -8.09 & -7.61 & -6.20 \end{Bmatrix} \tag{28a}$$

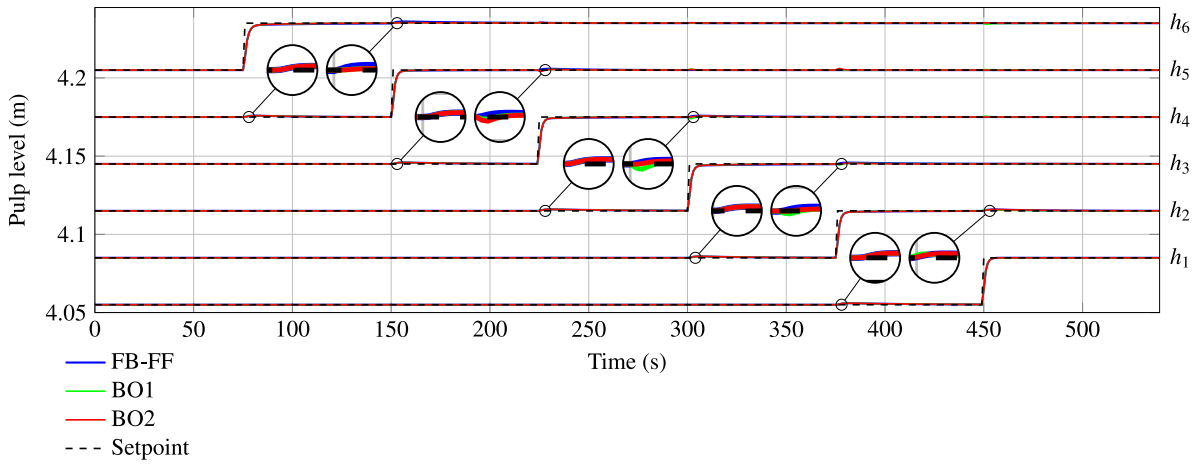

Fig. 5. Setpoint changes for Strategy 1 and 2 compared to FB-FF.

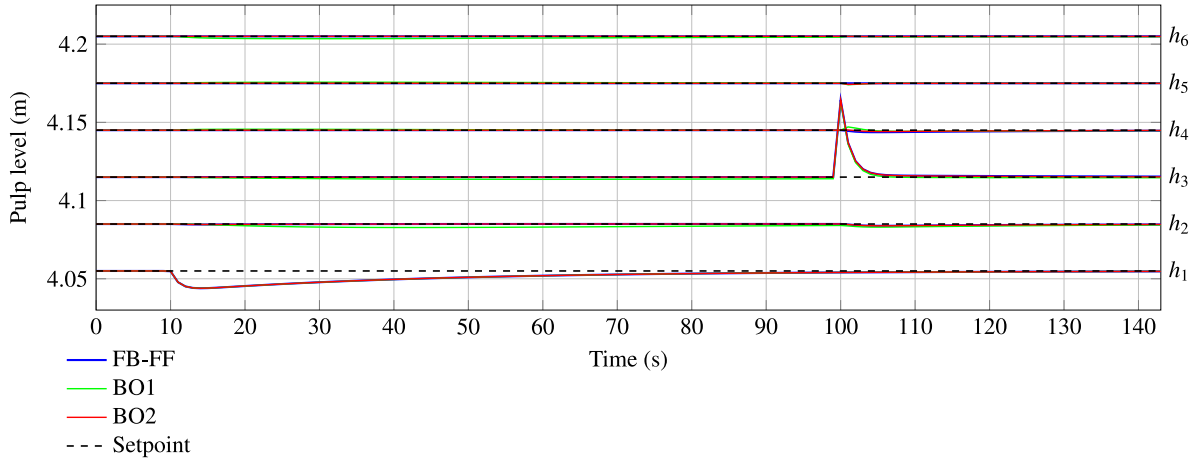

Fig. 6. Disturbance rejection for Strategy 1 and 2 compared to FB-FF.

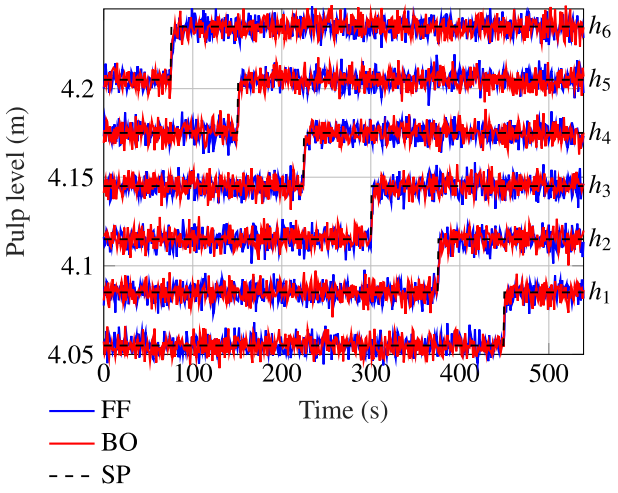

Fig. 7. Setpoint tracking performance with measurement noise.

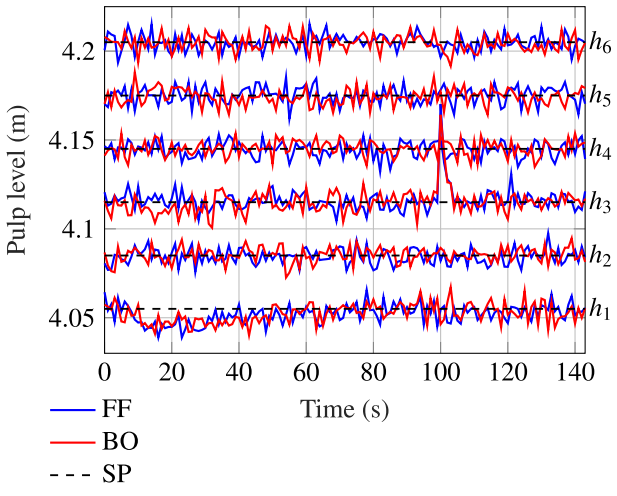

Fig. 8. Disturbance rejection with measurement noise.

$$\boldsymbol{\tau}_{Iij}^{(noise)} = \begin{Bmatrix} 0.579 & 0 & 0 & 0 & 0 & 0 \\ 0.438 & 0.409 & 0 & 0 & 0 & 0 \\ 0.213 & 0.197 & 0.538 & 0 & 0 & 0 \\ 0.435 & 0.304 & 0.394 & 0.444 & 0 & 0 \\ 0.837 & 0.266 & 1.401 & 0.874 & 0.338 & 0 \\ 0.805 & 0.418 & 0.612 & 0.392 & 0.388 & 0.565 \end{Bmatrix} \tag{28b}$$

It can be seen from both the controller gains and time constants that the BO automatic tuner favours detuning the controllers in the presence of noise. This makes sense as the original controller settings are quite aggressive and in the presence of noise it might be beneficial to detune

Table 7
Reference tracking performance with noise.

Metric	Controller		
	FB-FF	BO	Change
IAE	1120	1155	−3.13%
ISE	625	659	−5.44%
ITAE	301 077	314 759	−4.54%
Q	240 992	251 939	−4.54%

Table 8
Disturbance rejection performance with noise.

Metric	Controller		
	FB-FF	BO	Change
IAE	313	324	−3.51%
ISE	197	225	−14.0%
ITAE	22 506	22 776	−1.20%
Q	18 044	18 265	−1.22%

the controllers. The newly tuned controller performance is compared to the FB-FF controller. As seen in Tables 7 and 8 the controller given by the BO automatic tuner performs almost on par with the FB-FF controller.

The impact of the measurement noise can be seen in the change in the controller parameters and the error metrics from Tables 5 and 6 to Tables 7 and 8. This result is obtained without filtering the levels and shows the BO automatic tuner is capable of handling noisy environments while keeping the process stable. It does not provide an improvement as in the case with no noise, but does come close to equalling the performance of the FB-FF controller.

The use of filters on the level measurements might help in improving the controller settings found by the BO automatic tuner. However, in this work the goal is to test if the BO automatic tuner is able to function in a noisy environment, as filtering the level measurement signals might not get rid of all the noise and might come at the cost of information loss. Thus, the level measurement signals were not filtered in this work.

5.3. Iteration convergence and efficiency

The matters of iteration convergence and efficiency still remain. Convergence in this case implies that the level responses converge to an optimal response as defined in (24). As expected from the robust stability analysis in Section 4.2, the closed-loop responses for all iterations are stable as the BO only selects controller parameters that yield a stable closed-loop. The level responses of all 80 iterations for Strategy 2 for the first three columns of the noise free case are shown in Figs. 9 to 11. Here the iterations are plotted in grey, with the optimum, corresponding to the controller settings given in (26), plotted in red. As seen from the level responses all iterations return to setpoint, and thus the bounds found in Table 4 are valid and adhered to. (The BO iterations while tuning with measurement noise is not displayed as the added noise makes it very difficult to distinguish the different iterations from each other.)

Fig. 12 shows the objective function value of all 80 iterations for all columns. It is evident that much of the search space produce similar responses. The first column, where the most parameters (10) are being tuned, has the most variability.

The minimum objective function value at each iteration is shown in Fig. 13. A value of 1 represents a performance equal to that of the FB-FF controller. All the columns, except for column 1, show an improvement in performance. (Note, cell 1, which depends only on c_{11} in (22), does not show an improvement. As discussed in Section 4.1, this is expected since the diagonal elements are first tuned using BO, then fixed, and afterwards the off-diagonal elements are tuned, i.e., c_{11} is the same for both the FB-FF and the MIMO inventory controllers.) Fig. 13 shows that the iterations required to find the minimum of the objective function decreases as fewer parameters need to be tuned, i.e. as the automatic tuner moves further down the bank. Each of the 80 step tests is given 35 s to complete, which makes the total tuning time per column about 47 min and the total for the whole bank 3 h and 54 min.

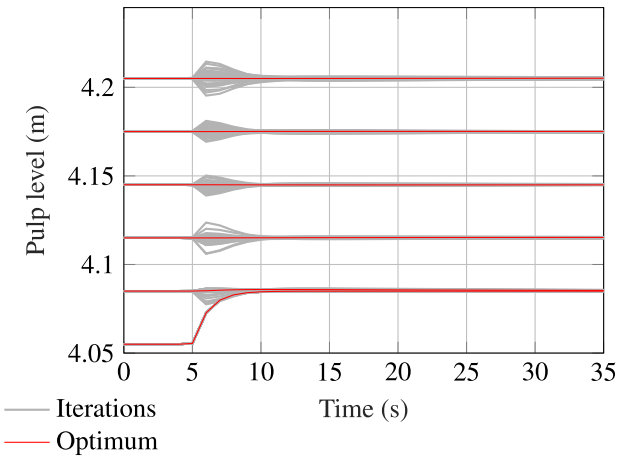

Fig. 9. Cell 1 BO iterations.

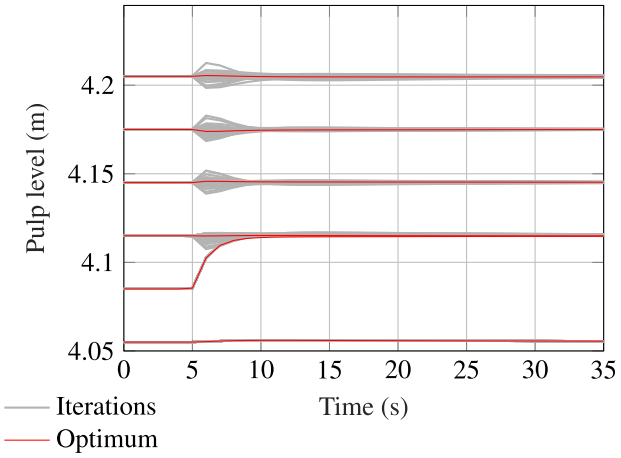

Fig. 10. Cell 2 BO iterations.

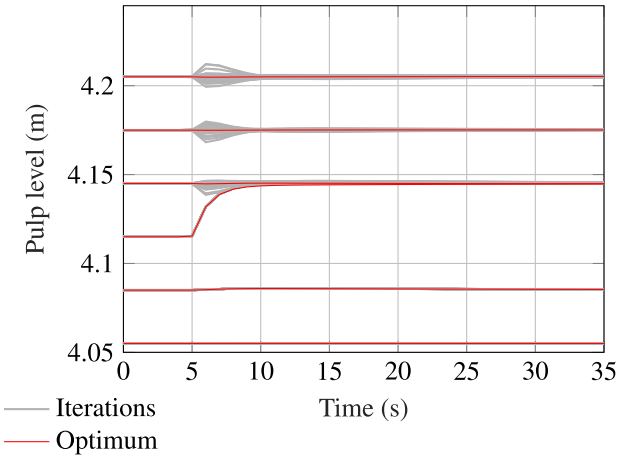

Fig. 11. Cell 3 BO iterations.

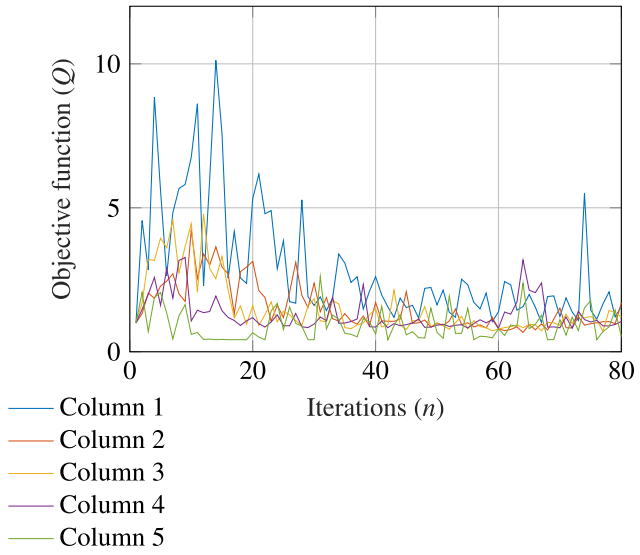

Fig. 12. Objective function value for each BO iteration.

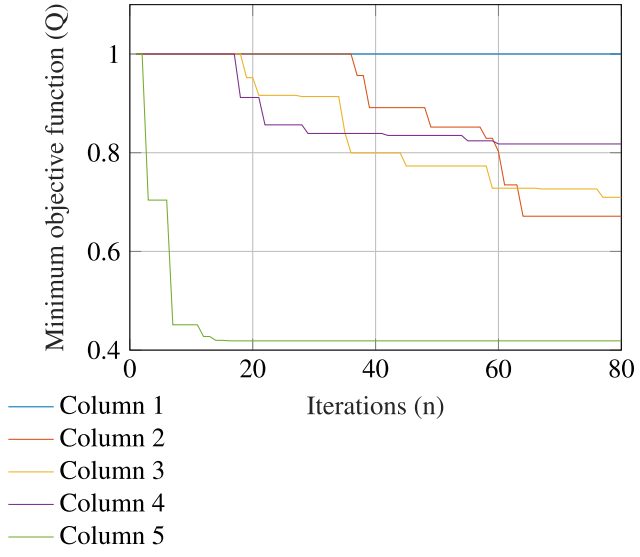

Fig. 13. Minimum of objective function value traces normalized to the performance of the FB-FF controller.

Table 9
Reference tracking performance for BO with different no. of iterations.

Metric	No. of Iterations		
	20	40	80
IAE	62.4	60.3	57.7
ISE	33.8	33.6	33.5
ITAE	17 629	17 312	16 274
Q	14 110	13 856	13 031
Change in Q from FB-FF	0.10%	1.89%	7.73%

The performance of the controllers obtained by the BO automatic tuner are compared for the cases where 20, 40 and 80 iterations are used respectively. The performance for each case is given in Tables 9 and 10. The BO tuner is able to obtain a larger improvement over the FB-FF controller with 80 iterations than with 20 or 40 iterations. In most the cases the minimum for each column is found well before the 80th iteration.

Table 10
Disturbance rejection performance for BO with different no. of iterations.

Metric	No. of Iterations		
	20	40	80
IAE	60.9	60.7	59.3
ISE	57.0	57.0	56.9
ITAE	3828	3775	3603
Q	3074	3031	2894
Change in Q from FB-FF	−1.82%	−0.41%	4.14%

Table 11
Circuit parameters and variables for 'industrial' case where there is model-plant mismatch.

τ_1	1.45	min	V_{cell_1}	63.6	m^3
τ_2	1.63	min	V_{cell_2}	66.1	m^3
τ_3	1.20	min	V_{cell_3}	66.83	m^3
τ_4	1.38	min	V_{cell_4}	68.99	m^3
τ_5	1.29	min	V_{cell_5}	67.17	m^3
τ_6	1.26	min	V_{cell_6}	71.21	m^3
B_1	12.05	$m^{2.5}/s$	C_{v_1}	1280	–
B_2	2.86	$m^{2.5}/s$	C_{v_2}	1182	–
B_3	19.82	$m^{2.5}/s$	C_{v_3}	1625	–
B_4	3.81	$m^{2.5}/s$	C_{v_4}	1458	–
B_5	5.26	$m^{2.5}/s$	C_{v_5}	1519	–
B_6	3.73	$m^{2.5}/s$	C_{v_6}	1649	–
Variable	Value	Unit	Variable	Value	Unit
h_1	3.87	m	h_4	4.30	m
h_2	4.61	m	h_5	4.04	m
h_3	3.49	m	h_6	4.31	m
Q_F	2569	m^3/s			

5.4. Model-plant mismatch for industrial case

In some cases it may be infeasible to allow the BO automatic tuner approximately 4 h to tune an industrial flotation bank. Since the circuit parameters in Table 1 and the level profile in Table 2 are available to plant operators or are relatively simple to calculate, it is possible to obtain a controller off-line in simulation similar to (28) and apply the controller obtained off-line to the industrial circuit. However, for various reasons there may be mismatch between the simulated model and industrial plant [36]. For example, silting may change the actual volume of a tank, the valves may be worn, the retention time may be different from what is expected, the feed flow-rate may have increased, or the level profile have changed. In this case, the BO automatic tuner can use the controller obtained off-line as the seed and be allowed a limited period to optimize the controller on the industrial circuit.

The simulation below investigates the case where a controller obtained off-line is applied to an 'industrial' circuit. The off-line controller in (28) is used as the seed controller on the industrial circuit which is further optimized by the BO automatic tuner. The industrial circuit in this case is simulated and represented by the set of parameters and variables as shown in Table 11 such that there is model-plant mismatch. The values were obtained by varying the parameters and variables in Tables 1 and 2 by a random percentage in the range $[-20, 20]$%. The BO automatic tuner was allowed 15 iterations (i.e., 44 min for the entire bank) to improve the seed controller and obtained the parameters in (29). No significant improvement is obtained after more iterations.

$$\mathbf{K}_{ij}^{noise} = \begin{Bmatrix} -8.08 & 0 & 0 & 0 & 0 & 0 \\ -4.22 & -8.08 & 0 & 0 & 0 & 0 \\ -3.69 & -3.00 & -7.66 & 0 & 0 & 0 \\ -4.38 & -5.31 & -3.34 & -8.08 & 0 & 0 \\ -3.38 & -2.53 & -2.15 & -6.16 & -8.08 & 0 \\ -2.39 & -2.36 & -5.93 & -6.09 & -3.71 & -6.20 \end{Bmatrix} \tag{29a}$$

Table 12
Reference tracking performance for industrial case. (Final columns show improvement of BO compared to other controllers.)

Metric	Controller				
	FB-FF	Seed	BO	% Improvement FB-FF	Seed
IAE	1725	1705	1696	1.68%	0.54%
ISE	1463	1439	1424	2.67%	1.02%
ITAE	474 909	467 799	459 202	3.31%	1.84%
Q	380 220	374 527	367 646	3.31%	1.84%

Table 13
Disturbance rejection performance for industrial case. (Final columns show improvement of BO compared to other controllers.)

Metric	Controller				
	FB-FF	Seed	BO	% Improvement FB-FF	Seed
IAE	578	564	519	10.21%	7.95%
ISE	674	629	556	17.51%	11.63%
ITAE	45 192	43 936	41 482	8.21%	5.59%
Q	36 288	35 275	33 297	8.24%	5.61%

$$\tau_{Iij}^{(noise)} = \begin{Bmatrix} 0.579 & 0 & 0 & 0 & 0 & 0 \\ 0.650 & 0.409 & 0 & 0 & 0 & 0 \\ 0.214 & 0.255 & 0.538 & 0 & 0 & 0 \\ 0.385 & 0.749 & 0.581 & 0.444 & 0 & 0 \\ 0.573 & 0.270 & 0.367 & 0.289 & 0.338 & 0 \\ 0.533 & 0.180 & 1.387 & 0.610 & 0.152 & 0.565 \end{Bmatrix} \quad (29b)$$

Tables 12 and 13 compare the performance of the seed controller to the final controller. Although the seed controller has reasonable performance, the BO automatic tuner can improve the controller performance to compensate for the model-plant mismatch in a relatively short period of time.

6. Conclusion

The BO automatic tuner is able to tune multiple PI elements to a satisfactory level to improve the overall performance in the level control of the flotation bank. A performance improvement is seen in the simulated level control for both the reference tracking and disturbance rejection. The BO automatic tuner successfully finds parameters that are on par or better than a FB-FF controller, even in the presence of measurement noise. The advantage in an industrial setting is that any flotation bank can be tuned automatically with relative ease, especially if a controller is first found off-line and the off-line controller is tuned further on-line for a limited time.

Future work can include exploring strategies for reducing the number of iterations required to achieve acceptable controller tuning parameters. Different perturbation methods could also be explored together with doing multiple rounds of tuning. Future work can also consider limiting the improvement that is searched for rather than just limiting the iterations and seeing how much improvement in the objective function can be obtained.

CRediT authorship contribution statement

Albertus V. Richter: Writing – original draft, Visualization, Validation, Methodology, Investigation. **Johan D. le Roux:** Writing – review & editing, Supervision, Conceptualization. **Ian K. Craig:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Number : 137769).

Data availability

No data was used for the research described in the article.

References

[1] L.G. Bergh, J.B. Yianatos, The long way toward multivariate predictive control of flotation processes, J. Process Control 21 (2) (2011) 226–234.
[2] B.J. Shean, J.J. Cilliers, A review of froth flotation control, Int. J. Miner. Process. 100 (3) (2011) 57–71.
[3] I. Jovanović, I. Miljanović, Contemporary advanced control techniques for flotation plants with mechanical flotation cells – A review, Miner. Eng. 70 (2015) 228–249.
[4] J.D. le Roux, D.J. Oosthuizen, S. Mantsho, I.K. Craig, A survey on the status of industrial flotation control, IFAC- Pap. 53 (2) (2020) 11854–11859.
[5] S. Skogestad, Control structure design for complete chemical plants, Comput. Chem. Eng. 28 (1) (2004) 219–234.
[6] L.E. Olivier, I.K. Craig, A survey on the degree of automation in the mineral processing industry, in: 2017 IEEE AFRICON, 2017, pp. 404–409.
[7] P. Kämpjärvi, S.-L. Jämsä-Jounela, Level control strategies for flotation cells, Miner. Eng. 16 (11) (2003) 1061–1068.
[8] K.J. Åström, T. Hägglund, Automatic tuning of simple regulators with specifications on phase and amplitude margins, Automatica 20 (5) (1984) 645–651.
[9] H. Boubertakh, M. Tadjine, P. Glorennec, S. Labiod, Tuning fuzzy PD and PI controllers using reinforcement learning, Int. Soc. Autom. Trans. 49 (4) (2010) 543–551.
[10] K.K. Ahn, D.Q. Truong, Online tuning fuzzy PID controller using robust extended Kalman filter, J. Process Control 19 (6) (2009) 1011–1023.
[11] R. Guardeño, M.J. López, V.M. Sánchez, MIMO PID controller tuning method for quadrotor based on LQR/LQG theory, Robotics 8 (2) (2019) 36.
[12] J.A. van Niekerk, J.D. le Roux, I.K. Craig, On-line automatic controller tuning using Bayesian optimisation - a bulk tailings treatment plant case study, IFAC-Pap. 55 (21) (2022) 126–131.
[13] S. Boyd, M. Hast, K.J. Åström, MIMO PID tuning via iterated LMI restriction, Internat. J. Robust Nonlinear Control 26 (8) (2015) 1718–1731.
[14] T.A.M. Euzébio, M.T. Da Silva, A.S. Yamashita, Decentralized PID controller tuning based on nonlinear optimization to minimize the disturbance effects in coupled loops, IEEE Access 9 (2021) 156857–156867.
[15] J.P.L. Coutinho, L.O. Santos, M.S. Reis, Bayesian optimization for automatic tuning of digital multi-loop PID controllers, Comput. Chem. Eng. 173 (2023) 108211.
[16] M. Fiducioso, S. Curi, B. Schumacher, M. Gwerder, A. Krause, Safe contextual Bayesian optimization for sustainable room temperature PID control tuning, 2019, arXiv preprint arXiv:1906.12086.
[17] H. Ono, T. Sonoda, A. Maekawa, Automatic tuning of PID controllers for MIMO processes, IFAC Proc. Vol. 33 (4) (2000) 79–84.
[18] F. Berkenkamp, A. Krause, A.P. Schoellig, Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics, Mach. Learn. 112 (10) (2021) 3713–3747.
[19] M. Neumann-Brosig, A. Marco, D. Schwarzmann, S. Trimpe, Data-efficient autotuning with Bayesian optimization: An industrial control study, IEEE Trans. Control Syst. Technol. 28 (3) (2020) 730–740.
[20] J.A. van Niekerk, J.D. le Roux, I.K. Craig, On-line automatic controller tuning of a multivariable grinding mill circuit using Bayesian optimisation, J. Process Control 128 (2023) 103008.
[21] S. Skogestad, I. Postlethwaite, Multivariable Feedback Control : Analysis and Design, John Wiley, Chichester ; Hoboken, NJ, 2007.
[22] A. Richter, J.D. Le Roux, I.K. Craig, Automatic tuning of level controllers in a flotation bank using Bayesian optimisation, IFAC- Pap. 58 (25) (2024) 13–18.
[23] J.T. Wilson, F. Hutter, M.P. Deisenroth, Maximizing acquisition functions for Bayesian optimization, Adv. Neural Inf. Process. Syst. 31 (2018).
[24] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. de Freitas, Taking the human out of the loop: A review of Bayesian optimization, Proc. IEEE 104 (1) (2016) 148–175.
[25] D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions, J. Global Optim. 13 (4) (1998) 455–492.
[26] S.-L. Jämsä-Jounela, M. Dietrich, K. Halmevaara, O. Tiili, Control of pulp levels in flotation cells, Control Eng. Pract. 11 (1) (2003) 73–81.
[27] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, J. Process Control 13 (4) (2003) 291–309.

[28] D. Hodouin, C. Bazin, E. Gagnon, F. Flament, Feedforward–feedback predictive control of a simulated flotation bank, Powder Technol. 108 (2) (2000) 173–179.
[29] P. Quintanilla, D. Navia, F. Moreno, S.J. Neethling, P.R. Brito-Parada, A methodology to implement a closed-loop feedback-feedforward level control in a laboratory-scale flotation bank using peristaltic pumps, MethodsX 10 (2023) 102081.
[30] B.A. Wills, J. Finch, Wills' Mineral Processing Technology: An Introduction to the Practical Aspects of Ore Treatment and Mineral Recovery, Butterworth-Heinemann, 2015.
[31] J.H. Schubert, R.G.D. Henning, D.G. Hulbert, I.K. Craig, Flotation control - a multivariable stabilizer, in: XIX International Mineral Processing Congress, San Francisco, Oct. 1995, pp. 237–244.
[32] E.M.B. Aske, S. Skogestad, Consistent inventory control, Ind. Eng. Chem. Res. 48 (24) (2009) 10892–10902.
[33] V. Minasidis, S. Skogestad, N. Kaistha, Simple rules for economic plantwide control, in: Krist V. Gernaey, Jakob K. Huusom, Rafiqul Gani (Eds.), Comput. Aided Chem. Eng. 37 (2015) 101–108.
[34] D.E. Seborg, D.A. Mellichamp, T.F. Edgar, F.J. Doyle, Process Dynamics and Control, John Wiley & Sons, 2019.
[35] VEGA, VEGAPULS 31, 2024, URL https://www.vega.com/en-za/products/product-catalog/level/radar/vegapuls-31. (Accessed 17 August 2024).
[36] L.E. Olivier, I.K. Craig, Development and application of a model-plant mismatch expression for linear time-invariant systems, J. Process Control 32 (2015) 77–86.