

Automatic Tuning of Level Controllers in a Flotation Bank using Bayesian Optimisation

Albertus V. Richter^{*,**} Johan D. Le Roux^{*,1} Ian K. Craig^{*}

^{*} *Department of Electrical, Electronic, and Computer Engineering,
University of Pretoria, Pretoria South Africa.*

^{**} *Mintek, Randburg South Africa.*

Abstract: A flotation bank consisting of 6 cells in series under Single-Input-Single-Output (SISO) Proportional Integral (PI) level control is automatically tuned using Bayesian Optimisation (BO). Open loop step tests from the valve position to the level are used to identify first-order plus time-delay (FOPTD) models for each flotation cell. The PI controller settings are tuned according to the Skogestad Internal Model Control (SIMC) tuning rules. Stability bounds derived from μ -analysis are defined using these SIMC settings. As the optimum achieved by the Bayesian optimiser is largely dependent on the parameter space provided to the tuning algorithm, this space is selected first to ensure stability and secondly for performance. The BO framework is able to tune each of the six SISO PI controllers to provide significantly improved level control over the original SIMC controllers with regards to different forms of the integrated error when the plant is subjected to step changes in the level setpoints and disturbances to the feed flow. This improvement comes at the cost of an increased number of tests to conduct.

Keywords: automatic tuning, Bayesian Optimisation, flotation level control, proportional and integral control, series tanks.

1. INTRODUCTION

The pulp levels of froth flotation banks are mainly controlled by Single-Input-Single-Output (SISO) Proportional and Integral (PI) controllers. The main focus of these base-level controllers is to keep the pulp level at a certain setpoint (Shean and Cilliers, 2011). The controller settings depend on the process dynamics of the plant. These settings are chosen to provide a characteristic/optimal response in terms of disturbance rejection and/or setpoint following requirements.

The flotation process dynamics on which the controller settings are based tend to change over time, with different ore being processed, different pulp level profiles and changes in valve and tank dynamics. Thus, the current controller settings become less optimal with the change in process dynamics. This change creates the need for re-tuning the PI controllers, which can be done in various ways. The classical tuning of PI controllers typically requires explicit system identification as seen from the step test methods (Cohen and Coon, 1953; Ziegler and Nichols, 1942). The classical relay excitation method introduced a more automated process of system identification (Åström and Hägglund, 1984; Schei, 1994; Voda and Landau, 1995). However, both the classical step test and relay excitation methods remove the controller from the loop to find controller settings, thus performing offline controller tuning. The classical methods have since been adapted to do online system identification with the controller in the loop to

try to improve the controller settings (Behera et al., 2017; Hang and Sin, 1991; Pavković et al., 2014; Tan et al., 2001).

The online nature of input-output data driven system identification (black box modelling) is of particular interest as it requires no structural change to the control loop for the identification. The data driven models can be obtained through various methods, such as Auto Regression Exogenous (ARX) and Implicit Model Reference Adaptive System (IMRAS) models (Behera et al., 2017; Yang et al., 2012). Reinforcement learning can be used to obtain implicit models (Wang et al., 2007; Howell and Best, 2000). Reinforcement learning seeks optimal controller settings that provide a minimum to a chosen cost function by iteratively adapting the controller parameters and evaluating the cost function. The iterative adaptation may require large data-sets and long process time for adequate performance. Fuzzy Logic Q-learning aims to reduce the learning time and data required for the learning process (Ahn and Truong, 2009; Boubertakh et al., 2010).

The use of Bayesian Optimisation (BO) seeks to optimise the search direction during optimisation of the cost function to reduce the number of iterations required to obtain an optimal parameter set (Berkenkamp et al., 2021). BO was successfully implemented in controller tuning for active disturbance rejection control (ADRC) in throttle valve control (Neumann-Brosig et al., 2020), as well as automatic tuning of Proportional, Integral and Derivative (PID) controllers used for room temperature control (Fiducioso et al., 2019). More recently, it was used to automatically tune mineral processing process units in

¹ Corresponding author. Email: derik.leroux@up.ac.za

simulation (van Niekerk et al., 2022, 2023). In this work, BO will be used to automatically tune PI level controllers in a flotation bank in simulation.

2. 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 minimise 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

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

where α is a vector of the parameters used to tune the given controller, in this case the proportional and integral parts of the PI controller. The cost function that is minimised by the BO algorithm defines the desired process response. The cost function for a SISO controller can be expressed as

$$Q = \sum_j^M w_j \beta_j(\alpha), \quad (2)$$

where β_j are the performance variables for $j = 1, \dots, M$ and w_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 tank.

3. BAYESIAN OPTIMISATION

The premise of BO is to obtain a surrogate model of how the cost function relates to the controller tuning parameters and find the minimum of this model within as few iterations as possible (Wilson et al., 2018). The surrogate model can be developed in a number of different ways. In this work, a Gaussian Process (GP) model will be used. An acquisition function is used to optimise the search direction to reduce the number of iterations needed to find the minimum of the cost function. The acquisition function is also used to propose the next set of parameters at which the cost function is evaluated. Numerous acquisition functions are available to be used depending on the application. The Expected Improvement (EI) function, which maximizes the information gained from a new parameter set, will be used in this work. With the two optimisation problems, BO consists of an inner problem where the acquisition function is used to maximise the search direction and an outer loop for the creation of the GP model and minimising the cost function. The structure of the BO algorithm can be seen below in Algorithm 1 from Shahriari et al. (2016).

Algorithm 1 : Bayesian Optimisation

```

for n = 1, 2, ..., pre-set value do
  → select new  $\alpha_{n+1}$  by optimising acquisition
  function  $\mathcal{L}$ :
   $\alpha_{n+1} = \arg \max_{\alpha \in \mathcal{A}} \mathcal{L}(\alpha | \mathcal{D})$ 
  → query objective function to obtain  $Q_{n+1}$ 
  → augment data  $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\alpha_{n+1}, Q_{n+1})\}$ 
  → update statistical model
end for

```

3.1 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 ($k(\alpha, \alpha')$) of the observed data points. The observed data points are collected in a data set \mathcal{D}

$$\mathcal{D} = \{\alpha_k, Q_k\}_k^N \quad (3)$$

where k is the new data point and N the total number of data points. 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 as

$$\text{cov}(Q(\alpha_p), Q(\alpha_q)) = k(\alpha_p, \alpha_q), \quad (4)$$

where the subscripts p and q denote different data points in the data set \mathcal{D} . The specific covariance function used is the standard *ARD Matern 3/2* function used in the MATLAB *bayesopt* function. The joint distribution of the prior is given by

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

where σ_n^2 is the variance of the training observations, \mathbf{Q} is the training observations at points A , and \mathbf{Q}_* is the predicted objective function at the test points A_* with $K(A, A)$, $K(A, A_*)$, $K(A_*, A)$ and $K(A_*, A_*)$ the covariance matrices obtained from the pairs of training data and test points. The predicted objective function at the test points is

$$\mathbf{Q}_* | A, \mathbf{Q}, A_* \sim \mathcal{N}(\bar{\mathbf{Q}}_*, \text{cov}(\mathbf{Q}_*)), \quad (6)$$

where

$$\bar{\mathbf{Q}}_* = K(A, A_*)^\top [K(A, A) + \sigma_n^2 I]^{-1} \mathbf{Q} \quad (7)$$

$$\begin{aligned} \text{cov}(\mathbf{Q}_*) &= K(A_*, A_*) \\ &\quad - K(A, A_*)^\top [K(A, A) + \sigma_n^2 I]^{-1} K(A, A_*). \end{aligned} \quad (8)$$

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

3.2 Acquisition function

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

$$\alpha_{n+1} = \arg \max_{\alpha \in \mathcal{A}} \mathcal{L}(\alpha | \mathcal{D}). \quad (9)$$

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

$$\mathbf{EI}(\alpha, Q) = \mathbb{E} \max[0, Q_{best}(\alpha_{best}) - Q(\alpha)], \quad (10)$$

where $Q_{best}(\alpha_{best})$ is the current best mean at the location α_{best} . This is solved analytically (Jones et al., 1998) as

$$\mathbf{EI}(\alpha) = (Q_{best} - \bar{\mathbf{Q}}_*(\alpha)) \Phi(z(\alpha)) + \sigma(\alpha) \phi(z(\alpha)), \quad (11)$$

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

$$z(\alpha) = \frac{Q_{best} - \bar{\mathbf{Q}}_*(\alpha)}{\sigma(\alpha)}, \quad (12)$$

where $\sigma(\alpha)$ is the predicted standard deviation at α .

4. FLOTATION PLANT MODEL

The flotation model used for the flotation bank with 6 cells in series can be found in Jämsä-Jounela et al. (2003). 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. 1.

The change in volume for each of the cells $i = 1, \dots, 6$ is given as

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

where C_v is the valve capacity coefficient, u_i is the valve position, h_i is the pulp level and H_i is the physical height difference between cells. The coefficient B_i is used to ensure a valve position of $u_i = 0.5$ in each cell at the initial flow rate (Q_F) from the grinding mill. The change in volume in the first cell is given by

$$\dot{V}_1 = Q_F - (B_1 C_v f_c(u_1)\sqrt{h_1 - h_2 + H_1}), \quad (14)$$

and the change in volume in the last cell by

$$\dot{V}_6 = (B_5 C_v f_c(u_5)\sqrt{h_5 - h_6 + H_6} - (B_6 C_v f_c(u_6)\sqrt{h_6 + H_6}). \quad (15)$$

The valve position $f_c(\cdot)$ is assumed to be a linear function of the control signal adapted around the steady state position, thus $f_c(u_i) = u_i + 0.5$, with $f_c(u_i) = 1$ being fully open and $f_c(u_i) = 0$ being fully closed. The physical height difference between cells is given by H_i and the pulp level in each cell by h_i . The change in pulp level is then given by

$$\dot{h}_i = \frac{\dot{V}_i}{A_i}, \quad (16)$$

where A_i is the cross-sectional area of the cell which is assumed to be cylindrical. The valve capacity coefficient C_v is given by

$$C_v = 1.17Q_m \sqrt{\frac{\rho_p}{\Delta p}}, \quad (17)$$

where $Q_m = 1.2 \frac{V_{cell}}{\tau/60}$ is the mean volume flow through a cell, $\Delta p = \rho_p g H_i$ is the pressure difference across a valve, ρ_p is the pulp density, and $g = 9.81 \text{ m/s}^2$ 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 cell level profile is given in Table 2.

Table 1. Flotation circuit parameters

Parameter	Value	Unit	Parameter	Value	Unit
Q_F	2336	m^3/h	$B_{i=1..5}$	3.49	$\text{m}^{2.5}/\text{s}$
V_{cell}	76.0	m^3	B_6	1.41	$\text{m}^{2.5}/\text{s}$
A_i	12.0	m^2	H_i	0.85	m
τ	1.50	min			

Table 2. Cell level profile

Parameter	Value	Unit	Parameter	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

5. CONTROLLER STRUCTURE AND TUNING BOUNDS

5.1 SISO PI control

The PI level controller for each cell is implemented as

$$u_i = u_{base} + K_c \left(1 + \frac{1}{\tau_I s}\right) \quad (18)$$

where the valve position (u_i) for each cell is the manipulated variable. The controller consists of the controller gain (K_c) and integral time constant (τ_I). The error signal ($e_i = h_i^{SP} - h_i$) is the difference between the level setpoint (h_i^{SP} or SP) and the measured pulp level (h_i) for each cell. The manipulated variable is adapted around a base point $u_{base} = 0.5$, thus the control action from the PI controller is limited to between -0.5 and 0.5 to keep the manipulated variable in the required range of 0 to 1.

The initial controller settings are found from open loop step tests where the valve positions are manipulated. The step responses are fitted to a first-order plus time-delay (FOPTD) approximation, $G(s) = \frac{k e^{-\theta s}}{\tau_p s + 1}$, where k is the system gain, τ_p is the time constant, and θ is the time delay. The obtained model parameters are shown in Table 3. All delays were obtained as 0, thus $\theta = 0$ for cells $i = 1, \dots, 6$. The controller settings are obtained using Skogestad's Internal Model Control (SIMC) as given in Table 4. The value for τ_c is chosen as $\tau_c = 0.1\tau_p$ (min) to provide a good trade-off between setpoint tracking and disturbance rejection (Seborg et al., 2019). The initial controller settings are given in Table 5.

5.2 Feedforward control

The level controllers make use of feedforward (FF) control to provide a degree of decoupling between cells. This is done by feeding forward the deviation of the valve position from nominal conditions to the next cell as seen in Kämpjärvi and Jämsä-Jounela (2003). The effects of this is that the PI controller of the next cell does not have to adapt to the changes in the flow generated by the control from the previous cell as this control is already added to the valve position by the FF control.

5.3 Controller bounds

The SIMC settings are used to define initial bounds for the auto-tuning procedure. The initial lower bound of the gain (K_c) parts of the controllers are set to a factor of 2 of the SIMC value. The initial upper bounds are selected in the opposite direction as a factor of 0.2 of the initial settings. The integral time-constant (τ_I) bounds are

Table 3. FOPTD model parameters

Cell	k	τ_p (min)	Cell	k	τ_p (min)
$i = 1$	-2.51	0.362	$i = 4$	-2.51	0.122
$i = 2$	-2.51	0.255	$i = 5$	-2.51	0.211
$i = 3$	-2.51	0.130	$i = 6$	-3.22	0.448

Table 4. SIMC tuning rules (Skogestad, 2003)

Model	$G(s)$	K_c	τ_I
FOPTD	$\frac{k e^{-\theta s}}{\tau_p s + 1}$	$\frac{1}{k} \frac{\tau_I}{\tau_c + \theta}$	$\min[\tau_p, 4(\tau_c + \theta)]$

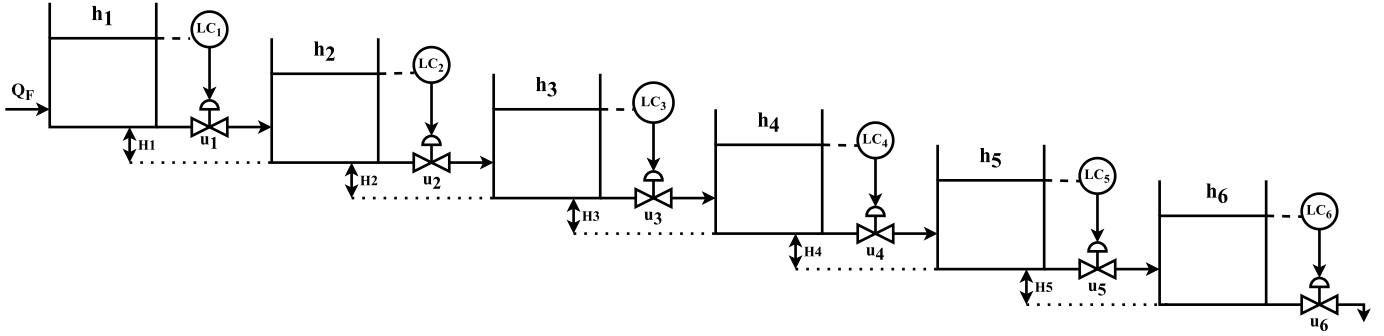


Fig. 1. Flotation bank configuration with PI level controllers for each cell.

Table 5. SIMC settings for each of the tanks

Cell	K_c	τ_I (min)	Cell	K_c	τ_I (min)
$i = 1$	-4.00	0.580	$i = 4$	-4.00	0.196
$i = 2$	-4.00	0.408	$i = 5$	-4.00	0.338
$i = 3$	-4.00	0.208	$i = 6$	-3.10	0.555

Table 6. SISO bounds for BO

Cell	K_c		τ_I (min)	
	Lower	Upper	Upper	Lower
$i = 1$	-8.980	-0.202	5.463	0.0616
$i = 2$	-8.980	-0.202	3.843	0.0433
$i = 3$	-8.980	-0.202	1.959	0.0221
$i = 4$	-8.980	-0.202	1.8460	0.0208
$i = 5$	-8.980	-0.202	3.183	0.0306
$i = 6$	-6.960	-0.157	5.227	0.0589

selected inversely to the proportional part, with the upper bound selected as a factor of 0.5 of the SIMC settings and a factor of 5 in the opposite direction. These bounds are subjected to a robust stability μ -analysis (Skogestad and Postlethwaite, 2007), where the minimal bounds for stability are found using the MATLAB Robust Control Toolbox. The bounds are given in Table 6.

The closed-loop system with individual PI controllers controlling each tank, is evaluated for stability robustness using the PI parameter bounds given in Table 6. The result can be visualised through the structured singular value (μ) plot of the system shown in Fig. 2. The μ -value is smaller than 1 for all frequencies implying the closed-loop is stable for all possible PI parameter values within the bounds provided. These bounds are subsequently used to bound the BO tuner.

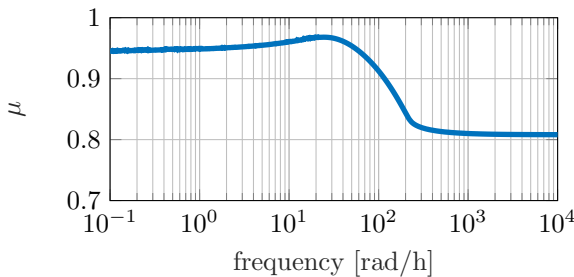


Fig. 2. Maximum structured singular values (μ) for the constraints given in Table 6.

6. SIMULATION

The SIMC controller settings and bounds are provided to the BO tuner. Closed-loop step tests are conducted one cell at a time starting with the first cell in the series. The BO is conducted on each cell using step tests to perturb the system and minimise the settling time of each cell level response to its own valve position. In other words, only a single cell is being tuned at a time. The BO iteratively changes the controller settings as discussed in Section 3 after each step test. The cost function in (2) returns the performance of the current settings. The returned cost function value is used in the acquisition function to suggest the next set of controller parameters. For this application the cost function is chosen as the settling time (T_s) of the cell level of the cell being tuned. This provides an almost critically damped response which also minimises the dynamics caused in the other cells by the step in the current cell. The cost function for a single cell is given by

$$Q_i = T_s(\alpha), \quad (19)$$

where α is the vector of the proportional and integral gain of the controller.

The SIMC and BO tuned controllers are evaluated according to different performance metrics: the Integral of the Absolute Error (IAE), the Integral Squared Error (ISE) and the Integral of the Time Absolute Error (ITAE) (Seborg et al., 2019). The plant is subjected to a sequence of setpoint changes as well as a feed flow disturbance in Q_F , combined with spillage water in cell 3. For the setpoint changes, a 3 cm increase in the level set-point of each cell is made, with the last cell being stepped first. The feed flow disturbance is a -20% step in Q_F and constant spillage water of 75 m³/h in cell 3 from time $t = 100$ s.

7. RESULTS

The BO tuner is allowed 20 iterations (i.e., 20 step tests with a duration 30 seconds each) to find the controller settings as this limit was found sufficient for the BO to reach the optimal value. These settings are applied to the

Table 7. Bayesian optimisation controller settings

Cell	K_c	τ_I (min)	Cell	K_c	τ_I (min)
$i = 1$	-8.08	0.852	$i = 4$	-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

PI controllers with and without FF control and compared to the performance of the SIMC settings with and without FF control. The BO settings are obtained without the FF control in the loop. As seen from Table 7 some of the optimal settings are found to be on the limit of the allowed boundaries, thus fast aggressive controllers are expected.

7.1 SISO SIMC and BO controllers without feedforward control

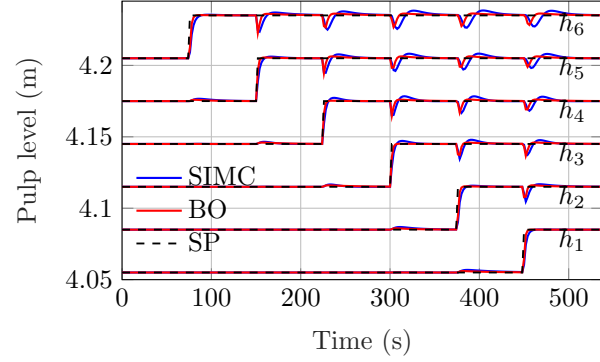


Fig. 3. Setpoint changes with PI only.

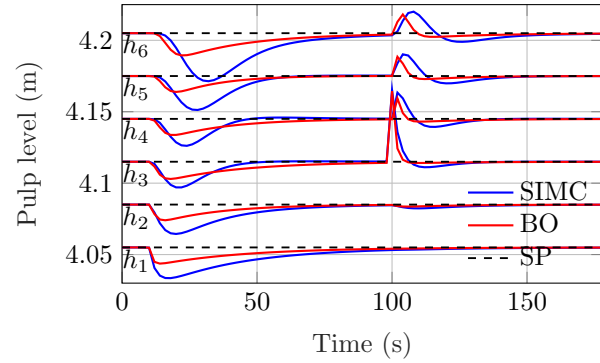


Fig. 4. Disturbance rejection with PI only.

The results for the SIMC and BO controllers without the use of FF control are shown in Figs. 3 and 4. The BO tuned PI controllers provide a significant improvement over the SIMC settings. The error peaks are reduced the most as can be seen from the ISE in Tables 8 and 9. The time the errors persist also shows improvement as seen in the ITAE. In both the setpoint tracking and disturbance rejection results the BO settings show a much more critically damped response compared to the SIMC settings, in line with minimising the settling time for each SISO PI controller. The faster, more aggressively tuned BO controllers reduce the effect of the interactions between the cells, reacting to the disturbances from the other cells much quicker.

Table 8. Setpoint change performance

Performance Metric	Controller		
	BO	SIMC	Improvement
IAE	1409	2714	48%
ISE	637	1491	57%
ITAE	468705	907043	48%

Table 9. Disturbance rejection performance

Performance Metric	Controller		
	BO	SIMC	Improvement
IAE	2652	4603	42%
ISE	1983	6684	70%
ITAE	132173	239699	44%

Table 10. Setpoint change performance with FF

Performance Metric	Controller		
	BO	SIMC	Improvement
IAE	659	1058	37%
ISE	294	655	55%
ITAE	178623	281258	36%

Table 11. Disturbance rejection performance with FF

Performance Metric	Controller		
	BO	SIMC	Improvement
IAE	591	1114	42%
ISE	428	1331	67%
ITAE	36631	64387	43%

7.2 SISO SIMC and BO controllers with feedforward control

The interactions between the cells can be further reduced with simple FF control as described in Section 5.2. The effect this has on the effectiveness of the controller settings is shown in the performance comparison of the SIMC and BO settings in both Tables 10 and 11, as well as Figs. 5 and 6. The BO settings still provide an improvement over the SIMC settings. The improvement is reduced from where no FF control was implemented. The error peaks are still the most improved as shown in the ISE, while the time taken to reduce the errors still shows significant improvement as per the ITAE. The results for the SIMC and BO controllers with FF control are shown in Figs. 5 and 6.

The improvement the FF control provides is best seen in the disturbance rejection of both SIMC and BO settings. The interactions between cells are greatly reduced, similar to the effect that can result from a well-designed multi-variable controller. Care should however be taken when using FF control to ensure that the plant model on which the controller is based is accurate. Otherwise FF control could degrade closed-loop performance (Seborg et al., 2019).

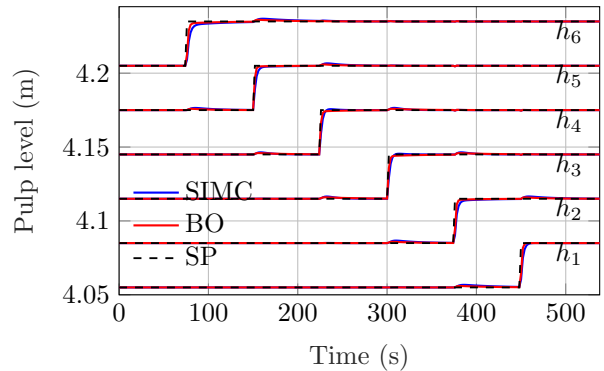


Fig. 5. Setpoint changes with PI and FF.

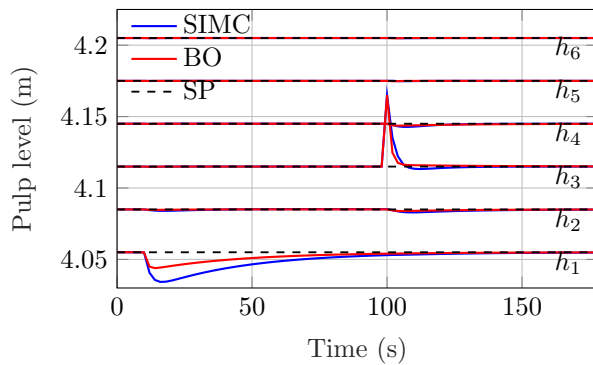


Fig. 6. Disturbance rejection with PI and FF.

8. CONCLUSION

The BO settings provide a significant improvement on the SIMC settings with regards to the ITAE, ISE and IAE performance metrics. The SIMC settings are used as a departure point and the bounded parameter space is searched to minimise the cost function to provide improved controller settings. This improvement comes at the cost of an increased number of step tests to conduct. The number of iterations required can be reduced in a number of ways, such as: exit functions, combined step tests and combined controller tuning. The latter is of interest as seen in the fact that additional control like the FF control can significantly improve the performance of the both SIMC and BO controllers. Future work includes considering automatic tuning of MIMO controllers using BO for the level controllers of a flotation bank.

REFERENCES

- Ahn, K. and Truong, D. (2009). Online tuning fuzzy PID controller using robust extended Kalman filter. *J. Process Contr.*, 19(6), 1011–1023.
- Åström, K. and Hägglund, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20(5), 645–651.
- Behera, S., Jyotirajan, M., and Pati, B. (2017). Optimal pole placement for a self tuning PID controller. *2017 6th Int. Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA)*, 456–461.
- Berkenkamp, F., Krause, A., and Schoellig, A.P. (2021). Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics. *Mach. Learn.*, 112(10), 3713–3747.
- Boubertakh, H., Tadjine, M., Glorennec, P.Y., and Labiod, S. (2010). Tuning fuzzy PD and PI controllers using reinforcement learning. *ISA Trans.*, 49(4), 543–551.
- Cohen, G.H. and Coon, G.A. (1953). Theoretical consideration of retarded control. *T. ASME*, 75, 827–834.
- Fiducioso, M., Curi, S., Schumacher, B., Gwerder, M., and Krause, A. (2019). Safe contextual Bayesian optimization for sustainable room temperature PID control tuning. *arXiv preprint arXiv:1906.12086*.
- Hang, C. and Sin, K. (1991). On-line auto tuning of PID controllers based on the cross-correlation technique. *IEEE T. Ind. Electron.*, 38(6), 428–437.
- Howell, M. and Best, M. (2000). On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata. *Contr. Eng. Pract.*, 8(2), 147–154.
- Jones, D.R., Schonlau, M., and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *J. Global Optim.*, 13(4), 455–492.
- Jämsä-Jounela, S.L., Dietrich, M., Halmevaara, K., and Tiili, O. (2003). Control of pulp levels in flotation cells. *Contr. Eng. Pract.*, 11(1), 73–81.
- Kämpjärvi, P. and Jämsä-Jounela, S.L. (2003). Level control strategies for flotation cells. *Miner. Eng.*, 16(11), 1061–1068.
- Neumann-Brosig, M., Marco, A., Schwarzmann, D., and Trimpe, S. (2020). Data-efficient autotuning with Bayesian optimization: An industrial control study. *IEEE T. Contr. Syst. T*, 28(3), 730–740.
- Pavković, D., Polak, S., and Zorc, D. (2014). PID controller auto-tuning based on process step response and damping optimum criterion. *Int. Society of Automation T.*, 53(1), 85–96.
- Schei, T.S. (1994). Automatic tuning of PID controllers based on transfer function estimation. *Automatica*, 30(12), 1983–1989.
- Seborg, D.E., Mellichamp, D.A., Edgar, T.F., and Doyle, F.J. (2019). *Process Dynamics and Control*. John Wiley & Sons.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., and de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.
- Shean, B. and Cilliers, J. (2011). A review of froth flotation control. *Int. J. Mineral Process.*, 100(3-4), 57–71.
- Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *J. Process Contr.*, 13(4), 291–309.
- Skogestad, S. and Postlethwaite, I. (2007). *Multivariable Feedback Control : Analysis and Design*. John Wiley, Chichester ; Hoboken, NJ.
- Tan, K., Lee, T., and Jiang, X. (2001). On-line relay identification, assessment and tuning of PID controller. *J. Process Contr.*, 11(5), 483–496.
- van Niekerk, J.A., le Roux, J.D., and Craig, I.K. (2022). On-line automatic controller tuning using Bayesian optimisation - a bulk tailings treatment plant case study. *IFAC-PapersOnLine*, 55(21), 126–131.
- van Niekerk, J.A., le Roux, J.D., and Craig, I.K. (2023). On-line automatic controller tuning of a multivariable grinding mill circuit using Bayesian optimisation. *J. Process Contr.*, 128, 103008.
- Voda, A. and Landau, I. (1995). A method for the auto-calibration of PID controllers. *Automatica*, 31(1), 41–53.
- Wang, X.s., Cheng, Y.h., and Sun, W. (2007). A proposal of adaptive PID controller based on reinforcement learning. *J. of China University of Mining and Technology*, 17(1), 40–44.
- Wilson, J.T., Hutter, F., and Deisenroth, M.P. (2018). Maximizing acquisition functions for bayesian optimization. *Adv. Neural Inf. Process. Syst.*, 31.
- Yang, Q., Xue, Y., Yang, S.X., and Yang, W. (2012). An auto-tuning method for dominant-pole placement using implicit model reference adaptive control technique. *J. Process Contr.*, 22(3), 519–526.
- Ziegler, J.G. and Nichols, N.B. (1942). Optimum settings for automatic controllers. *T. ASME*, 64(8), 759–768.