Journal of Process Control 128 (2023) 103008

Contents lists available at ScienceDirect

Journal of Process Control

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

On-line automatic controller tuning of a multivariable grinding mill circuit using Bayesian optimisation

J.A. van Niekerk [a,b], J.D. le Roux [b], I.K. Craig [b,*]

[a] Zutari, Pretoria, South Africa
[b] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa

ARTICLE INFO

Article history:
Received 14 November 2022
Received in revised form 26 April 2023
Accepted 29 May 2023
Available online 15 June 2023

Keywords:
Bayesian optimisation
Gaussian processes
Acquisition function
Auto-tuning
Ore milling

ABSTRACT

Process controllers are abundant in the industry and require attentive tuning to achieve optimal performance. While tuning controllers by the most primitive method of trial and error is possible, it often leads to sub-optimal performance if not conducted by a skilled expert. It is much more appealing to develop an on-line, sample efficient, automated tuner which can optimise the performance of a given controller to the task at hand. The automatic tuning procedure can be conducted during commissioning, when poor controller performance is observed or when process conditions have changed. The problem statement is formulated as the minimisation of an objective function constructed to achieve the desired controller performance. In this context the automatic tuning problem of multi-input multi-output (MIMO) controllers is considered within the framework of Bayesian optimisation and applied in simulation to an ore milling circuit with three manipulated and three controlled variables. Regulatory and set point tracking controllers are tuned automatically and are shown to achieve better performance than a reference controller.

1. Introduction

Process controllers and especially proportional integral derivative (PID) controllers are abundant in the industry. Although the use of model predictive control (MPC) is widespread [1], PID is by far the most common feedback controller due to its stability and simplicity. A survey of eleven thousand controllers in the continuous process industry indicated that 97% of those controllers implemented the PID algorithm [2]. PID controllers are also implemented extensively as part of decentralised controllers for multi-input multi-output (MIMO) processes [3].

Numerous PID tuning methods have been researched and published. Better known methods include Ziegler–Nichols [4], Cohen–Coon [5], IMC [6], SIMC [7], and AMIGO [8]. Luyben [9] describes the LACEY procedure which implements the biggest log-modulus tuning (BLT) method to tune decentralised PID controllers of MIMO processes with interaction between control loops.

In spite of the abundance of PID tuning methods available to the industry, Desborough and Miller [2] indicate that only a third of controllers provide an acceptable level of performance. This is partly due to the fact that the process of obtaining optimal tuning parameters can be expensive as it is time consuming to conduct system identification experiments which require the attention of domain experts and process downtime. Furthermore, frequent retuning may be required due to changing process conditions and ageing equipment. It is therefore evident that a need exists to optimally tune industrial controllers in an inexpensive manner. For that reason, this paper investigates the use of auto-tuners to optimally tune controllers.

Auto-tuning is not a novel concept and has enjoyed significant attention since the relay feedback method of Åström and Hägglund [10]. The relay feedback method was primarily intended to tune simple regulators of the PID type, and due to its success has subsequently received much research attention which has expanded its application to more complex controllers [11–17]. In light of the success of the relay feedback research, auto-tuners based on the relay feedback method have been commercialised [18].

Advancements made in computer processing capabilities and machine learning has provided an alternative approach to auto-tuning controllers by introducing self-learning techniques such as reinforcement learning [19]. Reinforcement learning has been used in auto-tuning applications since the turn of the century and application include the tuning of Ford Motors Zetec engines [20], a simulated inverted pendulum model [21], a CE150 helicopter model [21], flow rate of a desalination unit [22], and a non-linear tank system [23].

Neumann-Brosig et al. [24] considers the ideal auto-tuner to be on-line, model free, controller agnostic, data efficient and

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

https://doi.org/10.1016/j.jprocont.2023.103008

globally optimal (minimises the objective function). The ideal auto-tuner must interact with the live process, sample proposed tuning parameters, evaluate the performance of the proposal, and continue until the performance requirement of the controller has been optimised. Process sampling must be efficient and be conducted in as few as possible steps to minimise the expense of process downtime and production loss during performance evaluation.

This paper investigates Bayesian optimisation as a candidate for ideal auto-tuning of an industrial process. Bayesian optimisation is used to optimally tune a decentralised multivariable controller that controls an ore milling circuit in simulation. Such tuning can in theory be done on process models of ore-milling circuits that are available from the literature (e.g. [25]), but significant skill and effort are required to fit these models to an actual industrial process. Olivier and Craig [26] report that operators often do not understand the dynamics of the process and also do not know how to tune the controllers. Furthermore, Olivier and Craig [26] report that most grinding mill circuits are controlled using single-loop PID controllers for what is inherently a multivariable process (see also [27]). This is in contrast to the process industries in general where model-based controllers such as model predictive control dominate [28]. On-line, auto-tuners can be used to retune controllers when poor performance is observed due to plant-model mismatches that developed due to process disturbances. These factors make Bayesian optimisation an attractive alternative for the tuning of grinding mill circuit controllers.

The problem statement is defined as a fit for purpose objective function to be minimised by attentively selecting tuning parameters from a constrained search domain. Unlike reinforcement learning, Bayesian optimisation does not require off-line models for training and can be applied directly to the process.

Bayesian optimisation has been demonstrated to optimise the tuning paraments of a quadrotor vehicle [29]. Neumann-Brosig et al. [24] used Bayesian optimisation to find optimal tuning parameters of an active disturbance rejection controller (ADRC) for a throttle valve without the need for a process model and achieved better performance than trial-and-error tuning after only 10 experiments. Fiducioso et al. [30] used safe contextual Bayesian optimisation to optimise the PID parameters of a room temperature controller without human intervention. Lucchini et al. [31] and Sorourifar et al. [32] respectively applied Bayesian optimisation to tune MPCs for torque vectoring of high performance electrical vehicles and a continuously stirred tank reactor to notably improve performance. Lu et al. [33] shows than Bayesian optimisation with a reference model can effectively locate the global minimum in fewer iterations compared to traditional Bayesian optimisation methods. Paulson et al. [34] develops the adversarially robust Bayesian optimisation method for auto-tuning problems with significant time-invariant uncertainties.

Bayesian optimisation is well suited for the optimisation of unknown (i.e. black box) objective functions that are expensive to evaluate [35]. The expense can be expressed in any sense including computational effort, production down-time, cost of expertise or capital cost of evaluation. Snoek et al. [36] shows that Bayesian optimisation can reach or surpass human expert-level tuning of hyperparameters for machine learning algorithms. Lam et al. [37] applies Bayesian optimisation to address aerospace engineering applications where a finite budget of evaluations is available. The results of a black-box optimisation challenge held in 2020, demonstrates the benefits of Bayesian optimisation over random search methods for the tuning of hyperparameters [38].

This paper builds on the approach of Van Niekerk et al. [39] where Bayesian optimisation was applied to a tailings treatment plant. In this paper, Bayesian optimisation is applied to an ore milling circuit which is considered to be more challenging to control, given the increased dimensions of the plant and the stronger interactions between the manipulated and controlled variables. Van Niekerk et al. [39] implements an intuitive approach to determine the constraints of the search domain, whereas this paper applies robust stability analysis to analytically determine the constraints of the search domain. Not only does this novel approach increase the probability of including the global minimum of the objective function, but also eliminates the possibility of unstable iterations that could lead to personnel injury or equipment damage.

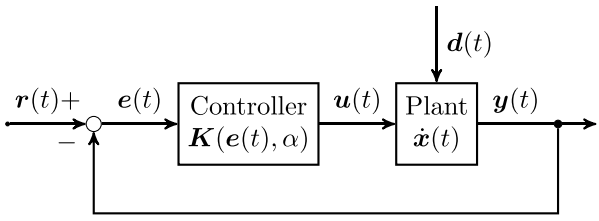

Fig. 1. Feedback controller.

The paper is structured as follows: Section 2 presents the problem statement of auto-tuning a MIMO process controller and the objective function to be optimised. Information on Bayesian optimisation, Gaussian processes and acquisition functions is provided in Section 3. Section 4 describes the process to be controlled, the approach to determining constraints and constructing objective functions. Minimisation of objective functions to achieve desired performance by means of Bayesian optimisation is demonstrated by simulation. The results are discussed in Section 5 and concluding remarks are provided in Section 6.

2. Problem statement

Consider a dynamic MIMO process of an industrial plant presented by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)) \tag{1a}$$

$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{d}(t)) \tag{1b}$$

that is to be controlled by a feedback controller where $\boldsymbol{x}(t)$ are the process states, $\boldsymbol{u}(t)$ are the manipulated variables, $\boldsymbol{y}(t)$ are the observed process variables and $\boldsymbol{d}(t)$ are the disturbances. The process model is not known *a priori*.

Assume that the plant of (1) is controlled by a controller in a unity feedback configuration as shown in Fig. 1. The controller can be represented by

$$\boldsymbol{u}(t) = \boldsymbol{K}(\boldsymbol{e}(t), \alpha) \tag{2}$$

were $\boldsymbol{K}$ is the controller, $\boldsymbol{e}(t)$ are the control errors which are the difference between the set points $\boldsymbol{r}(t)$ and the observed process variables, and α are the controller tuning parameters.

The tuning parameters of the controller that would provide optimum performance are unknown and must be sought. The tuning parameters $\alpha \in \mathcal{A}$ are constrained in the domain $\mathcal{A} \subseteq \mathbb{R}^d$. The performance of the tuning parameters are quantified in this work by evaluating each of the observed process variables in terms of time domain performance indices. These performance indices for controller evaluation could typically include the integral of squared error (ISE), integral of absolute error (IAE), integral of time multiplied squared error (ITSE), integral of time multiplied absolute error (ITAE), rise time, settling time, overshoot, output usage, decay ratio, etc. [40–42]. These indices are dependent on the inputs used with step changes being the most typical input variation.

Where multiple performance indices are used to evaluate a controller, they must be scaled according to the required response. The performance associated with each process variable is weighted and combined to provide an objective function representing the performance of the controller as a single scalar quantity. The objective function for a MIMO controller can be expressed as

$$Q = \sum_{i=1}^{n} \omega_i \left(\sum_{j=1}^{p} \beta_{ij} q_j(\boldsymbol{\alpha}) \right) \tag{3}$$

where Q is the objective function, n is the total number of process variables, ω_i is the process variable performance weighting, p is the total number of performance indices selected per process variable, q_j is the performance index and β_{ij} is a scaling factor to scale the contribution of each performance index. The selection of the weighting, performance index, and scaling factor will determine the performance of the optimised controller. These parameters remain fixed during the optimisation processes since the required performance of the optimised controller does not change as the iterations progress.

The form of the performance indices as functions of the tuning parameters is unknown, but can be calculated from experiments conducted on the process (1). Candidate tuning parameters are identified and selected for each experiment, the closed loop response is observed and the performance indices q_j and objective function Q are calculated. The experiments are performed iteratively, with a new set of tuning parameters selected for each iteration, until the global minimum of the objective function is found. The tuning of the controller can be expressed as a function to be optimised to find the set of tuning parameters that minimises

$$\min_{\alpha \in \mathcal{A}} = Q(\boldsymbol{\alpha}) \tag{4}$$

where $\boldsymbol{\alpha}$ is a vector consisting of the all tuning parameters as determined by the structure of the controller.

3. Bayesian optimisation

The Bayesian approach to optimisation is to first specify prior knowledge about the unknown objective function using a probabilistic surrogate model, and then to locate the global optimum of that model using an acquisition function [43]. Unlike random search and grid search optimisation techniques where past performance is not considered to locate the global optimum [44], Bayesian optimisation makes decisions based on the performance of previously sampled parameters. Such thoughtful choices of parameter selection characterise the sample-efficient nature of Bayesian optimisation [45].

The surrogate model is computationally cheaper to evaluate and optimise compared to an unknown objective function. The acquisition function evaluates the surrogate model to select the next set of parameters to be sampled on the objective function. In this way the cheap evaluation effort of the surrogate model is maximised while minimising the expensive evaluation effort of the objective function.

In this work the surrogate is modelled as a Gaussian process [46]. Gaussian processes not only provide predictions of unsampled inputs, but also the confidence of those predictions that can be interpreted in a natural way [47]. Several acquisition functions exist that can interpret Gaussian processes and identify the next input to be sampled. Compared to other surrogate models, Gaussian processes have a small number of training parameters [48]. The computational complexity of Gaussian processes increases cubically as the number of sampling points increase [49], but since it is an objective to limit the number of expensive experiments, this limitation is not of concern in this work.

3.1. Gaussian processes

Gaussian processes are described by their mean and covariance function and can be written as

$$Q(\boldsymbol{\alpha}) \sim \mathcal{GP}(m(\boldsymbol{\alpha}), k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')) \tag{5}$$

where $m(\boldsymbol{\alpha})$ is the mean function, which is normally taken to be zero for notational simplicity, and $k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ is the covariance function of $Q(\boldsymbol{\alpha})$. The covariance function is selected to capture prior knowledge about the shape of the objective function such as smoothness and rate of change. In contrast, the unrealistic smoothness of the commonly used squared exponential function makes it impractical for optimisation problems. To aid in the selection of the covariance function, Snoek et al. [36] propose the automatic relevance determination (ARD) Matérn parameter 5/2 kernel as the covariance function. This function is used in this work.

Gaussian processes learn the input–output relationships from a training dataset. For the problem statement defined in (3) and (4), the input is the tuning parameter vector $\boldsymbol{\alpha}$ and the output is the objective function value $Q(\boldsymbol{\alpha})$. Noisy observations can be modelled as

$$\widehat{Q} = Q(\boldsymbol{\alpha}) + \varepsilon \tag{6}$$

where $\widehat{Q}$ is the observed noisy objective function. The difference between the function value and observed value is due to additive noise assumed to have a Gaussian distribution with zero mean and variance σ_n^2

$$\varepsilon \sim \mathcal{N}(0, \sigma_n^2). \tag{7}$$

The inputs and outputs can be combined to form the training dataset $\mathcal{D} = \{(\boldsymbol{\alpha}_i, \widehat{Q}_i)|_{i=1}^n\}$ of n observations. Of primary interest is the knowledge gained about the function by incorporating the training dataset and prior distribution. The joint distribution of the observed function values and test outputs according to the prior is

$$\begin{bmatrix} \widehat{\boldsymbol{Q}} \\ \boldsymbol{Q}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(A,A) + \sigma_n^2 I & K(A,A_*) \\ K(A_*,A) & K(A_*,A_*) \end{bmatrix}\right) \tag{8}$$

where A denotes the design matrix consisting of all n inputs $\boldsymbol{\alpha}_i$ as column vectors. The observations $\widehat{Q}_i$ are collected in the column vector $\widehat{\boldsymbol{Q}}$ so that $\mathcal{D} = \{(A, \widehat{\boldsymbol{Q}})\}$. $\boldsymbol{Q}_*$ is the objective function prediction corresponding to test inputs A_* and $K(\cdot, \cdot)$ denotes covariances of the datapoints.

The predictive equations are obtained by deriving the conditional distribution from the joint distribution.

$$\boldsymbol{Q}_* | A, \widehat{\boldsymbol{Q}}, A_* \sim \mathcal{N}(\bar{\boldsymbol{Q}_*}, \mathrm{cov}(\boldsymbol{Q}_*)) \tag{9}$$

where,

$$\bar{\boldsymbol{Q}_*} = \boldsymbol{k}_*^\top [K + \sigma_n^2 I]^{-1} \widehat{\boldsymbol{Q}} \tag{10a}$$

$$\mathrm{cov}(\boldsymbol{Q}_*) = \boldsymbol{k}_{**} - \boldsymbol{k}_*^\top [K + \sigma_n^2 I]^{-1} \boldsymbol{k}_*. \tag{10b}$$

$\bar{\boldsymbol{Q}_*}$ is the mean prediction and the variance is the diagonal elements of $\mathrm{cov}(\boldsymbol{Q}_*)$. The compact notations are $K = K(A,A)$, $\boldsymbol{k}_{**} = K(A_*,A_*)$ and $\boldsymbol{k}_* = K(A,A_*)$.

3.2. Acquisition function

In Bayesian optimisation, acquisition functions are used to search the parameter space to acquire the next input location to be sampled based on the predictive mean and variance of the surrogate objective function. The objective of the acquisition function is not to learn the entire unknown objective function, but only to locate the global minimum (or maximum, depending on the objective function) within the constraints provided [50].

The acquisition function balances the trade-off between exploration and exploitation. Focussing the search to where the predictive mean is low promotes exploitation while searching where the variance is high favours exploration [50].

Acquisition functions identify the next input location to be sampled by finding the point where the acquisition function $\mathcal{L}$ is maximised, with [36]

$$\boldsymbol{\alpha}_* = \underset{\boldsymbol{\alpha}\in\mathcal{A}}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\alpha}|\mathcal{D}) \tag{11}$$

where $\boldsymbol{\alpha}_*$ is the next input location to be sampled given the training dataset $\mathcal{D}$.

Acquisition functions that can interpret Gaussian processes include amongst other, expected improvement (EI), Gaussian process upper confidence bound, and probability of improvement [36]. In this work EI [51] is selected, as it has been shown to escape local optimums [52], is better behaved than probability of improvement, and does not require a tuning parameter such as the Gaussian process upper confidence bound [36].

EI is the maximum expected improvement over the current best input location and is defined as

$$\mathrm{EI}(\boldsymbol{\alpha}) = \mathbb{E}\max[0, \widehat{Q}(\boldsymbol{\alpha}_{min}) - \widehat{Q}(\boldsymbol{\alpha})] \tag{12}$$

where $\boldsymbol{\alpha}_{min}$ is the location of the current best (minimum) posterior mean. When the posterior distribution is Gaussian, EI can be solved analytically [53] as

$$\mathrm{EI}(\boldsymbol{\alpha}) = \begin{cases} \Psi(\boldsymbol{\alpha}) + \Phi(\boldsymbol{\alpha}), & \text{if } \sigma(\boldsymbol{\alpha}) > 1 \\ 0, & \text{if } \sigma(\boldsymbol{\alpha}) = 0 \end{cases} \tag{13}$$

where,

$$\Psi(\boldsymbol{\alpha}) = (\widehat{Q}(\boldsymbol{\alpha}_{min}) - \bar{\mathbf{Q}}_*(\boldsymbol{\alpha}))\psi(Z) \tag{14a}$$

$$\Phi(\boldsymbol{\alpha}) = \sigma(\boldsymbol{\alpha})\phi(Z) \tag{14b}$$

$$Z = \frac{(\widehat{Q}(\boldsymbol{\alpha}_{min}) - \bar{\mathbf{Q}}_*(\boldsymbol{\alpha}))}{\sigma(\boldsymbol{\alpha})}. \tag{14c}$$

$\sigma(\boldsymbol{\alpha})$ is the predicted standard deviation at $\boldsymbol{\alpha}$, ϕ and ψ denote the probability density function (PDF) and cumulative distribution function (CDF) of the normal distribution respectively. Eq. (12) is differentiable and can be maximised with a gradient based optimiser to obtain $\boldsymbol{\alpha}_*$.

Algorithm 1: Bayesian optimisation

1: **for** n = 1,2, …,pre-set value **do**
2: select new $\boldsymbol{\alpha}_*$ by maximising acquisition function $\mathcal{L}$
$\boldsymbol{\alpha}_* = \operatorname{argmax}_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{L}(\boldsymbol{\alpha}|\mathcal{D}_n)$
3: sample process at $\boldsymbol{\alpha}_*$ to observe $\widehat{Q}_{n+1}$
4: augment data set $\mathcal{D}_{n+1} = \{D_n, (\boldsymbol{\alpha}_*, \widehat{Q}_{n+1})\}$
5: update posterior distribution
6: **end for**

Bayesian optimisation is a cyclic process that progresses as follows:

- The acquisition function identifies the next input location $\boldsymbol{\alpha}_*$ to be sampled.
- The process is sampled by means of an on-line experiment at $\boldsymbol{\alpha}_*$.
- The next input location $\boldsymbol{\alpha}_*$ to be sampled and the observed objective function $\widehat{Q}_{n+1}$ of the on-line experiment is appended to the previous training dataset to create a new augmented dataset $\mathcal{D}$.
- The posterior distribution (surrogate model) of the objective function is updated using the augmented dataset.

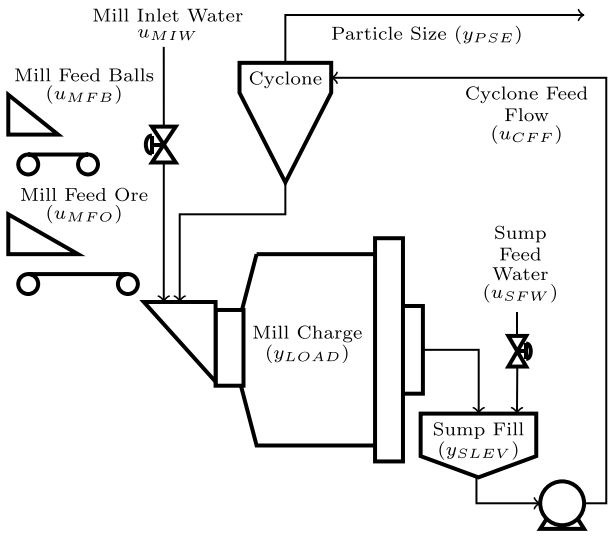

Fig. 2. ROM ore milling circuit.

This process repeats itself until a predetermined number of cycles has been reached. Refer to Algorithm 1 for the pseudocode of the process [50]. The global minimum is $\min(\widehat{\boldsymbol{Q}})$, which is the minimum of all the observations accumulated and not necessarily the result of the last acquisition cycle.

To apply Bayesian optimisation to a process, as in the case study below, it is necessary to define the objective function and the constraints for the search domain. Whereas the objective function and any associated weights and scaling parameters are process specific, a robust stability analysis can be used to obtain the constraints for the search domain.

4. Simulation

4.1. Plant

The controller selected to optimise by means of Bayesian optimisation is the controller for a run of mine (ROM) ore milling circuit. A brief introduction of the process is provided here.

Fig. 2 illustrates the process flow of a milling circuit with single stage classification. The semi-autogenous (SAG) mill is fed with ROM ore, water and steel balls. The steel balls are normally batched by an operator but for the purposes of this study are assumed to be continuously fed. Slurry is discharged from the mill through a screen and collected in the sump where it is diluted with sump feed water. The aperture of the screen determines the particle size distribution of the discharge slurry. The slurry is pumped to a hydrocyclone for classification were the lighter particles that are within specification overflow to the downstream process. The heavier out of specification particles are returned via the cyclone underflow to the mill for further grinding.

Table 1 lists the milling circuit variables of interest. The controlled variables are the fraction of the mill filled with charge (y_{LOAD}), the level of the slurry in the sump (y_{SLEV}) and the fraction of particles in the product with a size smaller than 75 μm (y_{PSE}). The manipulated variables are the feed rate of ore to the mill (u_{MFO}), feed rate of dilution water to the sump (u_{SFW}) and feed rate of diluted slurry to the cyclone (u_{CFF}). The feed rate of water to the mill (u_{MIW}) can be used to extend the control range of y_{PSE} [54], but for the purposes of this study will not be used as a manipulated variable. Instead, it will be set at a constant flow rate. The variable constraints and steady state operating points are listed in Table 2. At an ore feed rate of $u_{MFO} = 100$ t/h, the mill is filled to $y_{LOAD} = 0.45$ and the sump level is maintained at $y_{SLEV} = 5$ m^3 to provide a product with $y_{PSE} = 0.8$ [25,55,56].

Table 1
Variable descriptions.

Variable	Description
Controlled variables	
y_{SLEV}	Sump level [m^3]
y_{PSE}	Product particle size estimate [fraction < 75 μm]
y_{LOAD}	Total charge in the mill [fraction]
Manipulated variables	
u_{CFF}	Cyclone feed flow rate [m^3/h]
u_{SFW}	Sump feed flow rate [m^3/h]
u_{MFO}	Mill feed ore [t/h]

Table 2
Variable constraints and operating point.

Variable	Min	Max	Operating point
y_{SLEV}	2	9.5	5
y_{PSE}	0.6	0.9	0.8
y_{LOAD}	0.3	0.5	0.45
u_{CFF}	400	500	443
u_{SFW}	0	400	267
u_{MFO}	0	200	100

4.2. Optimisation of set point tracking

Consider the scenario where the milling circuit is controlled by decentralised controllers tuned using the SIMC method [7] during a desktop study prior to commissioning. The SIMC method requires a linear model to design the controllers. Such a linear model (15) is derived from the non-linear model from Coetzee et al. [56] and Le Roux et al. [25] and is presented in the form

$$\boldsymbol{y} = \boldsymbol{G}_p(s)\boldsymbol{u} + \boldsymbol{G}_d(s)d \tag{15}$$

where

$$\boldsymbol{y} = \left[y_{SLEV}, y_{PSE}, y_{LOAD}\right]^T \tag{16}$$

$$\boldsymbol{u} = \left[u_{CFF}, u_{SFW}, u_{MFO}\right]^T. \tag{17}$$

The plant transfer function is

$$\boldsymbol{G}_p(s) = \begin{bmatrix} g_{p11} & g_{p12} & g_{p13} \\ g_{p21} & g_{p22} & g_{p23} \\ g_{p31} & g_{p32} & g_{p33} \end{bmatrix} \tag{18}$$

where

$$g_{p11} = \frac{-0.29}{s} \tag{19a}$$

$$g_{p12} = \frac{0.42}{s} \tag{19b}$$

$$g_{p13} = 0 \tag{19c}$$

$$g_{p21} = \frac{-0.00035(1-0.63s)}{(1+0.54s)}e^{-0.011s} \tag{19d}$$

$$g_{p22} = \frac{0.0055}{1+0.24s}e^{-0.011s} \tag{19e}$$

$$g_{p23} = \frac{-0.0043}{1+0.58s}e^{-0.065s} \tag{19f}$$

$$g_{p31} = \frac{0.0028(1+0.876s)}{(1+3.868s)}e^{-0.0115s} \tag{19g}$$

$$g_{p32} = 0 \tag{19h}$$

$$g_{p33} = \frac{0.01}{s}. \tag{19i}$$

The disturbance transfer function is

$$\boldsymbol{G}_d(s) = \begin{bmatrix} \frac{-0.24}{(1+0.54s)}e^{-0.014s} \\ \frac{1.86\times10^{-3}}{(1+14.9s)}e^{-0.438s} \\ \frac{0.58}{(1+1.41s)}e^{-0.089s} \end{bmatrix} \tag{20}$$

where the disturbance $d = \eta_f$, and η_f is the hardness of the ore expressed in terms of power per ton of fines produced η_f [kWh/t].

The steady state relative gain array (RGA) [57] suggest input–output pairings of $u_{CFF} - y_{SLEV}$, $u_{SFW} - y_{PSE}$, and $u_{MFO} - y_{LOAD}$ for decentralised control. With the absence of second order terms in the transfer function, derivative action can be omitted, and the decentralised controller $\boldsymbol{K}_\alpha$ can be structured with only PI controllers on the diagonal.

$$\boldsymbol{K}_\alpha = \begin{bmatrix} k_{11} & 0 & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & k_{33} \end{bmatrix} \tag{21}$$

The PI controllers in Laplace domain are of the form [41]

$$k_{jj} = k_{Pjj}(1 + \frac{1}{\tau_{Ijj}s}), j = 1, 2 \tag{22}$$

where k_P is the proportional gain and τ_I is the integral time constant measured in hours. By applying the SIMC tuning rules of first-order and integrating processes, to an assumed diagonal plant, the PI controller parameters are calculated to be

$$k_{P11} = -22.989,\ \tau_{I11} = 0.6 \tag{23a}$$

$$k_{P22} = 206.807,\ \tau_{I22} = 0.24 \tag{23b}$$

$$k_{P33} = 500,\ \tau_{I33} = 0.8. \tag{23c}$$

Consider that post commissioning, as part of production performance evaluation, closed-loop set point step tests are conducted on the full model of (18) using the controller $\boldsymbol{K}_\alpha$ defined by (21) through (23) to gauge the performance. The results captured in Figs. 8 and 9 show that the response of y_{PSE} is overdamped with a settling time of 1.47 h. The response of y_{LOAD} is underdamped with a peak overshoot of approximately 14% and a settling time of 2.09 h. The performance may be considered inadequate due to plant-model mismatch, equipment replacement or deterioration of equipment performance. Improving the set point tracking ability of the controller is required to benefit the supervisory layer of a production or economic optimiser [58,59]. To this end, Bayesian optimisation is applied as an on-line and automated tuner to retune the controller for improved set point tracking. To implement Bayesian optimisation, constraints must be set and a suitable objective function selected as discussed in the following sections.

4.2.1. Constraints

Bayesian optimisation is a constrained regression process, and the constraints must be considered with care. The constraints determine the domain within which the Bayesian optimisation algorithm must search for the optimal tuning parameters to minimise the objective function. The search domain must be sufficiently large to include the optimum, but also restricted to prevent unstable iterations.

To expand the search space around the tuning parameters of the known controller $\boldsymbol{K}_\alpha$, a robust stability analysis [40,60] is conducted on an initial set of constraints to determine how much uncertainty over and above the initial constraints can be tolerated. The initial gain constraints are cautiously selected as a factor of 2 in the direction of instability, and boldly selected as

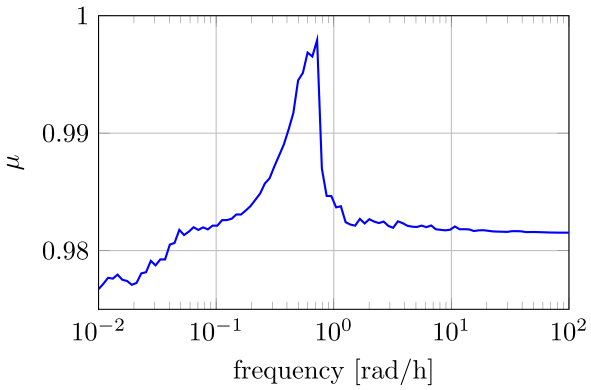

Fig. 3. Robust stability structured singular value (μ) plot with tuning parameters constrained as per (24).

a factor of 0.2 in the opposite direction. Selection of the initial integral time constraints follows the opposite approach, i.e. a factor of 0.5 in the direction of instability and a factor of 5 in the opposite direction. The Robust Control Toolbox of MATLAB provides the stability margins for the uncertain system incorporating controllers with uncertain tuning parameters. The robust stability analysis provides the maximum parameter uncertainty that can be tolerated before the worst-case uncertainty yields instability. The maximum parameter uncertainty determines the constraints of the search domain, which are

$$k_{P11} \in [-52.69, -0.539] \tag{24a}$$

$$\tau_{I11} \in [0.262, 25.55] \tag{24b}$$

$$k_{P22} \in [4.413, 473.2] \tag{24c}$$

$$\tau_{I22} \in [0.105, 10.22] \tag{24d}$$

$$k_{P33} \in [11.71, 1146] \tag{24e}$$

$$\tau_{I33} \in [0.349, 34.07]. \tag{24f}$$

By expanding the search domain to the threshold of instability as given in (24), the probability of including the optimal tuning parameters to find the global minimum of the objective function is increased. While inclusion of the global optimum cannot be guaranteed because the robust stability analysis is conducted on a linear system model, it is likely that the global minimum is contained in the search space. By excluding parameters that yield instability from the search domain, exploration can take place on-line without exposing operators or equipment to unsafe operations caused by unstable tuning parameters".

Fig. 3 shows the robust stability μ plot with the tuning parameter search domain constrained as per (24). Since $\mu < 1$ for all frequencies, it confirms that the closed-loop transfer function of $\boldsymbol{G}_p$ and $\boldsymbol{K}_\alpha$, with the parameter ranges of (24), will remain stable during Bayesian optimisation. In addition, the μ values are close to 1 over the frequency range of interest, as they should be given that parameter ranges of (24) represent the closed-loop system at the threshold of instability.

4.3. Objective function for set point tracking

The objective function selected to retune controller $\boldsymbol{K}_\alpha$ for improved set point tracking, presented in the form of (3), is

$$Q_{track} = \omega_1\beta_{11}q_1 + \omega_2\beta_{22}q_2. \tag{25}$$

Q_{track} consists of two terms. The first term represents evaluation of y_{PSE} and the second the evaluation of y_{LOAD}. Each term requires a step test to evaluate and therefore each Bayesian optimisation iteration will consist of two step tests, e.g., y_{PSE} is stepped and observed before y_{LOAD} is stepped and observed. The performance weights ω_1 and ω_2 are selected to penalise a particular output to promote a favourable response. Both outputs are considered to be of equal importance and therefore the weights are chosen as $\omega_1 = \omega_2 = 1$.

The sump acts as a buffer to absorb disturbances and a regulatory controller should aim to keep the sump from overflowing or running dry. Improvement of the sump level set point tracking performance will carry the overhead of an additional step test with no economic benefit. Evaluation of the sump level performance is therefore excluded from the set point tracking objective function.

Performance index q_1 is the ITAE of y_{PSE} and q_2 is the ITAE of y_{LOAD} in response to a set point step change. Using ITAE as the performance index has the benefit that the objective function value can be calculated without having to wait for all the transient dynamics to die out, which reduces the evaluation period of each iteration. A further beneficial property of ITAE is that it penalises both the absolute error as well as the persistence of the error making it useful for set point tracking evaluation. Figs. 8 and 9 show that the transient dynamics of $\boldsymbol{K}_\alpha$ have mostly decayed after 2 h which is therefore selected as the evaluation period.

Settling time (time taken for the error to stay within 2% of $|y_{final} - y_{initial}|$) was also evaluated as a candidate performance index. Using settling time as performance index has the benefit that no scaling factor is required since both responses will be measured against the same time scale with comparable magnitudes. The drawback of settling time is that the evaluation period must be long enough to allow the response of the candidate tuning parameters to settle. Should the response not settle within the provided evaluation period, the settling time cannot be measured, the objective function cannot quantify the performance and the result of the iteration does not contribute towards the training dataset $\mathcal{D}$, i.e., it is a wasted iteration. Due to the large integral time and small proportional gain parameter values included in the search space, the response of the slower controllers takes more than 20 h to settle. Evaluation periods of 20 h per step tests are impractical if step tests require production down time while there are other options to consider such as ITAE.

The inconvenience of using ITAE is that the terms are not similar in magnitude and need to be scaled. The scaling factors used in (25) are $\beta_{11} = \frac{1}{0.914}$ and $\beta_{22} = \frac{1}{0.75}$. The scaling factors are the inverse ITAE values in response to set point step changes of controller $\boldsymbol{K}_\alpha$ integrated over a 2 h period which requires a step test to calculate.

4.4. Optimisation of disturbance rejection

Consider the scenario where the milling circuit feed is sourced from ore stockpiles with different physical properties such as hardness and size distribution. The varying physical properties manifest as disturbances which may lead to solids hold-up in the mill, fluctuations in the circulating load and inconsistent product particle sizes [61]. Karageorgos et al. [62] describes the general trend towards a reduction of surge capacity and the need to maintain stability regardless of disturbances.

Hardness and size distribution are known to be correlated, i.e., the harder the ore the coarser the feed. Therefore the only disturbance considered is ore hardness [61].

Bayesian optimisation is applied to retune controller $\boldsymbol{K}_\alpha$ for improved disturbance rejection. The search domain constraints remain the same as for set point tracking but a fit for purpose objective function is required.

4.5. Objective function for disturbance rejection

The objective function selected to retune controller $\boldsymbol{K}_\alpha$ for improved disturbance rejection, presented in the form of (3), is

$$Q_{reject} = \omega_1\beta_{11}q_1 + \omega_2\beta_{22}q_2 + \omega_3\beta_{33}q_3. \quad (26)$$

Q_{reject} consists of three terms representing y_{SLEV}, y_{PSE} and y_{LOAD} respectively. The weights are selected to be $\omega_1 = \omega_2 = \omega_3 = 1$ since the disturbance rejection of all three outputs are considered to be of equal importance.

Performance index q_1 is the ITAE of the sump level, q_2 is the ITAE of the y_{PSE}, and q_3 is the ITAE of y_{LOAD} in response to an ore hardness step change. Performance indices q_1, q_2 and q_3 are calculated from a single step test per iteration, which reduces the integration period compared to an objective function requiring multiple step tests.

The ITAE scaling factors in (26) are $\beta_{11} = \frac{1}{12353}$, $\beta_{22} = \frac{1}{380}$ and $\beta_{33} = \frac{1}{511}$. The scaling factors are the inverse ITAE values in response to a disturbance step change of controller $\boldsymbol{K}_\alpha$ integrated over an 8 h period.

5. Results

Closed-loop step tests are conducted on the MIMO plant model by stepping set points or disturbances. For the case of evaluating set point step changes, the objective function requires two step tests. For the case of disturbance step changes, a single step is sufficient for each Bayesian optimisation iteration since the objective function is constructed for a single disturbance input.

5.1. Set point tracking

Figs. 4 and 5 show how the controlled variables and manipulated variables respond to a y_{PSE} set point step change and how Bayesian optimisation explores the search space by applying candidate tuning parameters to minimise the objective function. The best iteration is highlighted and represents the response of controller $\boldsymbol{K}_{track}$. $\boldsymbol{K}_{track}$ is the best result of optimising $\boldsymbol{K}_\alpha$ in (21) by minimising Q_{track} in (25). The y_{PSE} set point is stepped from a fraction of 0.8 to 0.9. Control of y_{PSE} is paired with u_{SFW} and therefore u_{SFW} immediately increases in response to the increased y_{PSE} demand. y_{SLEV} rises due to the increased u_{SFW} and as a result u_{CFF} increases to prevent the sump from overflowing. Interaction between u_{CFF} and y_{LOAD} causes y_{LOAD} to surge. u_{MFO} is throttled to recover from the increased y_{LOAD} and returns y_{LOAD} to the operating point. During the iteration process the controlled and manipulated variables all remain within operational bounds by limiting the size of the set point step change and constraining the search domain to robust stability margins.

Figs. 6 and 7 show how the controlled variables and manipulated variables respond to a set point step change in y_{LOAD}. The y_{LOAD} set point is stepped from a fraction of 0.45 to 0.5. Control of y_{LOAD} is paired with u_{MFO} and therefore u_{MFO} immediately increases in response to the increased demand in y_{LOAD}. y_{PSE} decreases due to the increased u_{MFO} and as a result the u_{SFW} increases to stabilise the y_{PSE}. The increased u_{SFW} causes y_{SLEV} to rise, and u_{CFF} is increased to prevent the sump from overflowing. During the iteration process the controlled and manipulated variables all remain within operational bounds.

Figs. 8 and 9 show the response of y_{PSE} and y_{LOAD} to step change and compares the tracking performance of controller $\boldsymbol{K}_\alpha$ and the controller retuned using Bayesian optimisation $\boldsymbol{K}_{track}$. Objective function (25), selected to improve set point tracking, can be seen to improve the y_{PSE} settling time from 1.47 to 0.32 h. The y_{LOAD} settling time is reduced from 2.09 to 0.22 h and the

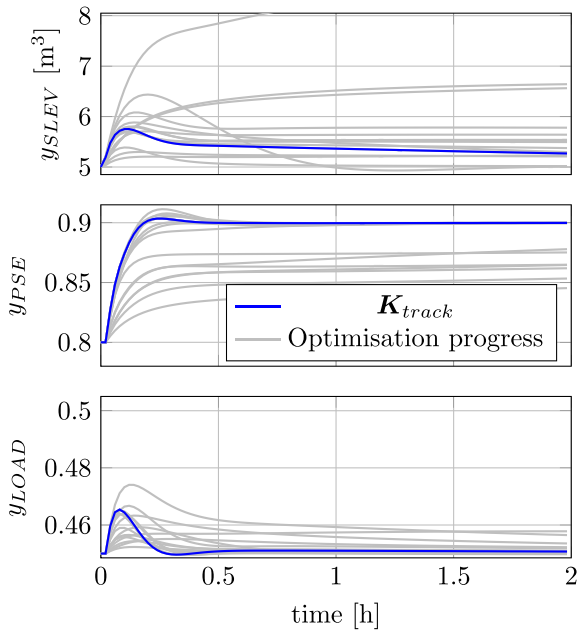

Fig. 4. Response of the controlled variables to a y_{PSE} set point step change during Bayesian optimisation using objective function Q_{track}.

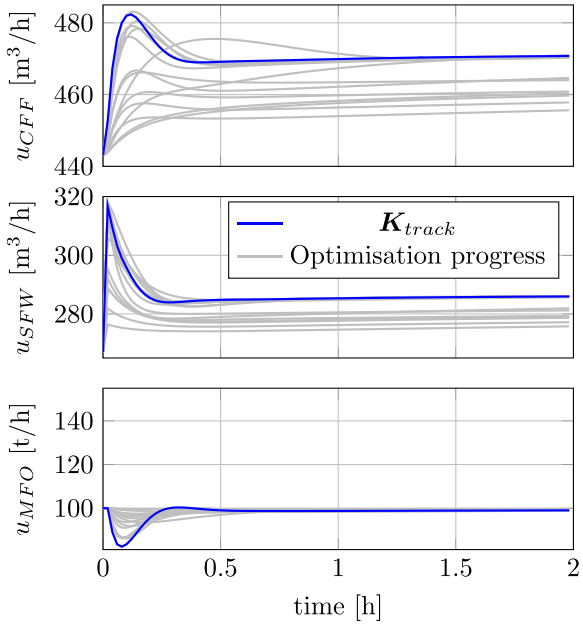

Fig. 5. Response of the manipulated variables to a y_{PSE} set point step change during Bayesian optimisation using objective function Q_{track}.

peak amplitude reduced from a fraction of 0.507 to 0.5. Table 3 lists the ITAE value reduction which is the basis of objective function (25). It provides a statistical evaluation comparing the root mean square error (RMSE), and compares the settling time of the controllers. The ITAE and RMSE values are calculated over a 2 h period.

Table 4 shows the results of iterations 6 through to 15 of the Bayesian optimisation simulation using objective function (25). Column Q_{track} represents the objective function value for each set of tuning parameters evaluated. During simulation, the step test response is evaluated over a period of 2 h. With each Bayesian iteration requiring two step tests, the 15 iterations as suggested in Table 4 will require no less than 60 h to complete in practice. The

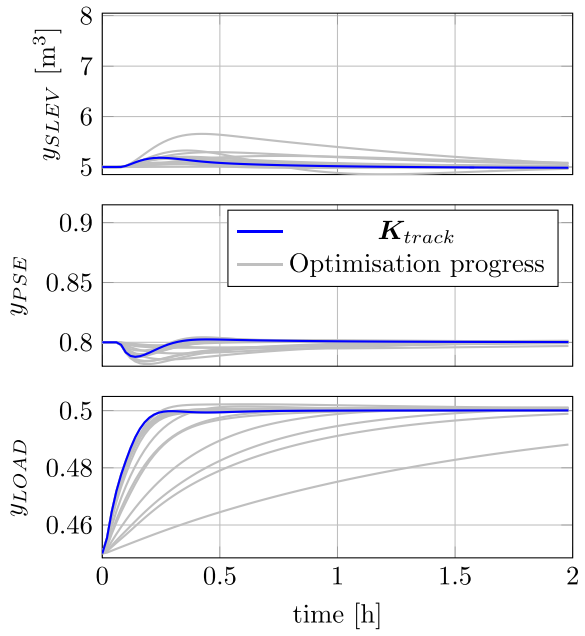

Fig. 6. Response of the controlled variables to a y_{LOAD} set point step change during Bayesian optimisation using objective function Q_{track}.

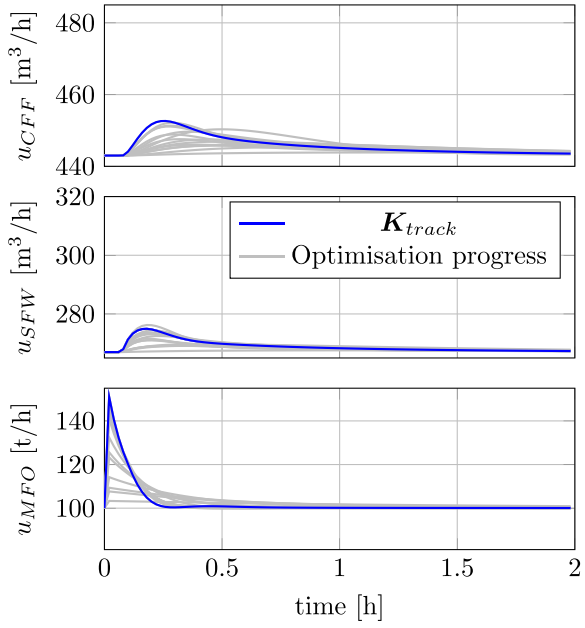

Fig. 7. Response of the manipulated variables to a y_{LOAD} set point step change during Bayesian optimisation using objective function Q_{track}.

Table 3
Comparison of set point tracking properties of controllers $\boldsymbol{K}_{track}$ and $\boldsymbol{K}_{\alpha}$. The improvement that controller $\boldsymbol{K}_{track}$ offers is indicated as a percentage.

Performance	$\boldsymbol{K}_{track}$	$\boldsymbol{K}_{\alpha}$	Impr. (%)
$y_{PSE\ ITAE}$	0.21	0.914	77.1
$y_{LOAD\ ITAE}$	0.123	0.75	83.7
$y_{PSE\ RMSE}$	0.030	0.044	31.5
$y_{LOAD\ RMSE}$	0.015	0.021	28.0
$y_{PSE\ SettlingTime}$	0.32	1.47	86.7
$y_{LOAD\ SettlingTime}$	0.22	2.09	89.5

best result is found by iteration 13. The results achieved as shown in Figs. 8 and 9 are satisfactory and conducting further iterations

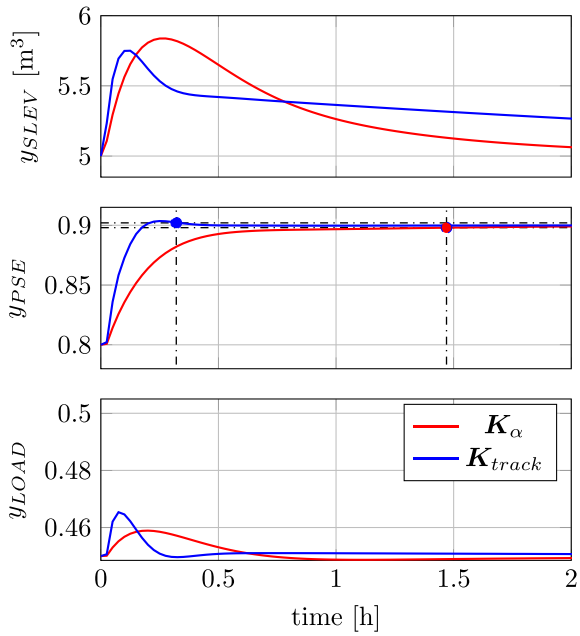

Fig. 8. Comparison of the set point tracking performance of controllers $\boldsymbol{K}_{track}$ and $\boldsymbol{K}_{\alpha}$ in response to a y_{PSE} set point step change. The markers indicate the settling time of the responses.

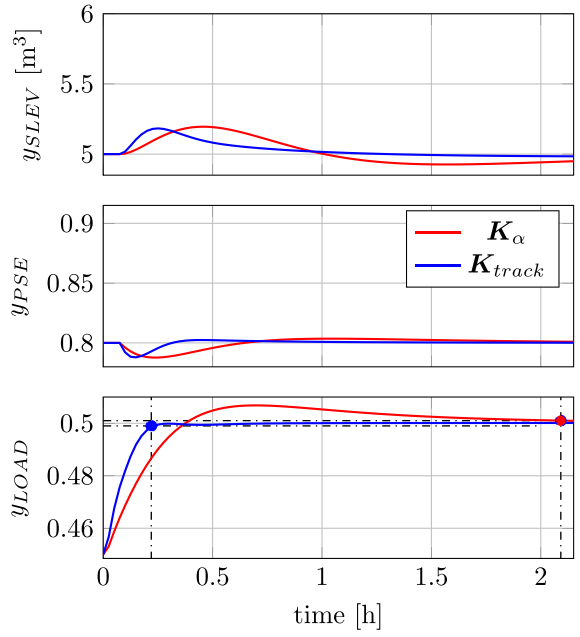

Fig. 9. Comparison of the set point tracking performance of controllers $\boldsymbol{K}_{track}$ and $\boldsymbol{K}_{\alpha}$ in response to a y_{LOAD} set point step change. The markers indicate the settling time of the responses.

in search of the global minimum at an overhead of 4 h per iteration does not warrant any further increase in performance.

The tuning parameters corresponding to the best iteration are

$$k_{P11} = -50.772,\ \tau_{I11} = 2.741 \tag{27a}$$

$$k_{P22} = 466.81,\ \tau_{I22} = 0.154 \tag{27b}$$

$$k_{P33} = 1144.2,\ \tau_{I33} = 21.105. \tag{27c}$$

Table 4
Results of Bayesian optimisation simulation using objective function (25), iterations 6 through 15.

Iteration	Q_{track}
6	7.767
7	0.881
8	0.511
9	6.714
10	0.665
11	2.164
12	0.492
13	**0.393**
14	1.047
15	0.558

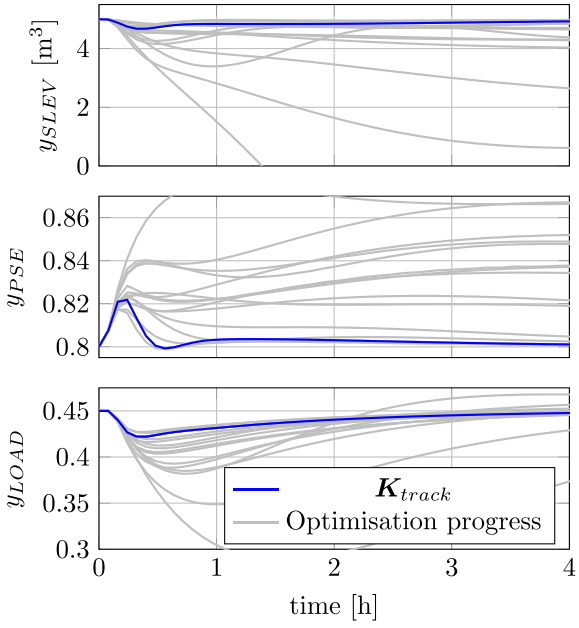

Fig. 10. Response of the controlled variables to an ore hardness step change during Bayesian optimisation using objective function Q_{reject}.

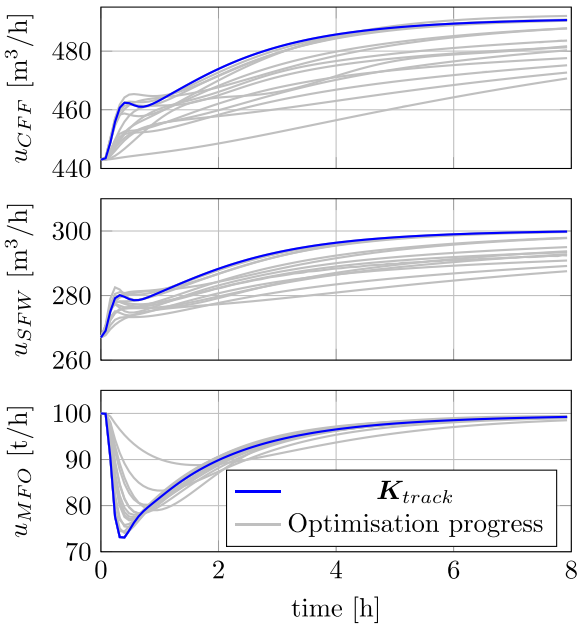

Fig. 11. Response of the manipulated variables to an ore hardness step change during Bayesian optimisation using objective function Q_{reject}.

Table 5
Comparison of disturbance rejection properties of controllers $\boldsymbol{K}_{reject}$ and $\boldsymbol{K}_{\alpha}$. The improvement that controller $\boldsymbol{K}_{reject}$ offers is indicated as a percentage.

Performance	$\boldsymbol{K}_{reject}$	$\boldsymbol{K}_{\alpha}$	Impr. (%)
$y_{SLEV\ ITAE}$	7083.2	12353	42.7
$y_{PSE\ ITAE}$	154.1	380.7	59.5
$y_{LOAD\ ITAE}$	461.4	511.7	9.85
$y_{SLEV\ RMSE}$	0.115	0.2	42.5
$y_{PSE\ RMSE}$	0.004	0.007	47.1
$y_{LOAD\ RMSE}$	0.01	0.011	13.0
$y_{SLEV\ Peak}$	0.331	0.517	36.0
$y_{PSE\ Peak}$	0.023	0.030	22.7
$y_{LOAD\ Peak}$	0.028	0.040	30.1

5.2. Disturbance rejection

Figs. 10 and 11 show how the controlled variables and manipulated variables respond to a 2.5% reduction in ore hardness and how Bayesian optimisation explores the search space by applying candidate tuning parameters to minimise the objective function. $\boldsymbol{K}_{reject}$ is the best result of optimising $\boldsymbol{K}_{\alpha}$ in (21) by minimising Q_{reject} in (26). The reduction in ore hardness causes an increase of y_{PSE} and reduction of y_{SLEV} and y_{LOAD}. The controller reacts by increasing the u_{SFW} and u_{CFF}. u_{MFO} drops to counter the effect of reduced ore hardness before returning to the initial feed rate. The manipulated variables do not saturate during the optimisation process. The sump is shown to run dry during one of the iterations. u_{CFF} is close to the maximum limit indicating that the plant and decentralised controller will not be able to cater for ore hardness disturbances much greater than 2.5% before u_{CFF} saturates and the sump overflows.

Fig. 12 compares the disturbance response of $\boldsymbol{K}_{reject}$ and $\boldsymbol{K}_{\alpha}$ to a step change in the feed ore hardness. The objective function performance criteria were selected to minimise the ITAE of the response and as a beneficial consequence the absolute error and the persistence of the error too. The ITAE values of the controlled variable responses are listed in Table 5 and shows how Bayesian optimisation brought about the reduction of 42.7%, 59.5% and 9.85% for the ITAE values of y_{SLEV}, y_{PSE} and y_{LOAD} respectively. The y_{LOAD} ITAE shows a significantly smaller improvement compared to the y_{SLEV} and y_{PSE} ITAE improvement. The peak disturbance of y_{SLEV}, y_{PSE} and y_{LOAD} improved by 36%, 22.7% and 30.1% respectively. The ITAE and RMSE values are calculated over an 8 h period.

While the ITAE and peak performance criteria showed good improvement, the comparatively poor performance of y_{LOAD} could be improved by adjustment to the objective function to penalise the y_{LOAD} ITAE. Simulation showed that doubling the y_{LOAD} performance weight and rescaling the objective function with ITAE values from Table 5 led to a significant improvement in the y_{LOAD} ITAE at the expense of the y_{PSE} ITAE. A consistent y_{PSE} has shown to result in better downstream product recovery and therefore improving y_{LOAD} disturbance rejection in favour of y_{PSE} disturbance rejection was not pursued.

From Fig. 12 it is evident that the transients due to disturbances take much longer to decay compared to the set point step changes of Figs. 8 and 9. The transient times (time it takes for the error to stay within to 2% of the peak error) for y_{SLEV}, y_{PSE} and y_{LOAD} are 8.4, 6.7 and 7.0 h respectively. Transient time differs from settling time in that transient time is a function of the maximum error caused by the disturbance while settling time is a function of the output change ($|y_{final} - y_{initial}|$) in response to a set point step change.

Simulations show that evaluation periods of up to 24 h are required for the objective function to provide a useful training

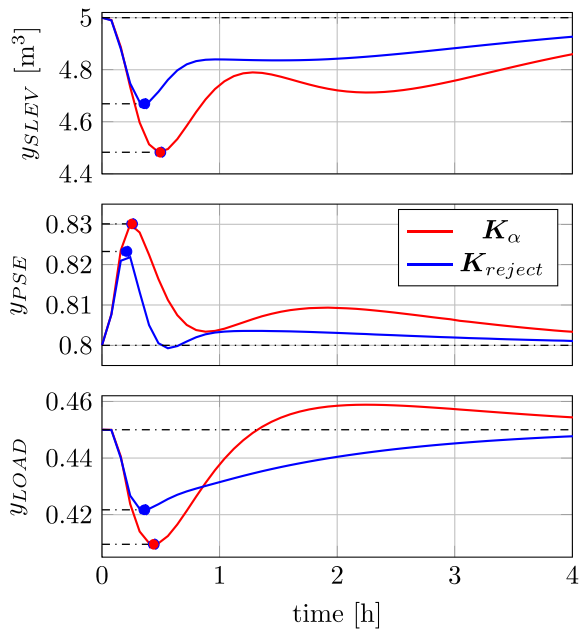

Fig. 12. Comparison of the disturbance rejection performance of controllers $\boldsymbol{K}_{reject}$ and $\boldsymbol{K}_\alpha$ in response to an ore hardness step change. The markers indicate the peak disturbance error of the responses.

Table 6
Results of Bayesian optimisation simulation using objective function (26), iterations 6 through 15.

Iteration	Q_{reject}
6	0.049
7	0.087
8	0.035
9	0.136
10	0.023
11	0.056
12	**0.013**
13	0.017
14	0.44
15	0.085

dataset $\mathcal{D}$ making transient time an unsuitable performance index for disturbance rejection.

Table 6 shows the results of iterations 6 through to 15 of the Bayesian optimisation simulation using objective function (26). During simulation, the step test response was evaluated over a period of 4 h to calculate the ITAE. From Fig. 12 it can be seen that the disturbance peaks have decayed after 4 h. With each Bayesian optimisation iteration only requiring a single step, the 15 iterations as suggested in Table 6 would require no less than 60 h to complete in practice. The best result is found by iteration 13. Note that the iterations do not stop once the global minimum is located but continues until the pre-set number of 15 iterations are complete.

The tuning parameters corresponding to the best iteration are

$$k_{P11} = -45.607,\ \tau_{I11} = 0.684 \tag{28a}$$

$$k_{P22} = 247.97,\ \tau_{I22} = 0.115 \tag{28b}$$

$$k_{P33} = 956.8,\ \tau_{I33} = 32.947. \tag{28c}$$

6. Conclusion

Despite the abundance and diverse range of control technologies at the disposal of the process control engineer, PID controllers are indispensable in the control hierarchy due to their simplicity and practicality. In spite of their benefits, frequent re-tuning may be required due to changing process conditions. This work demonstrates that Bayesian optimisation is a data efficient, on-line tuning method that can locate optimal tuning parameters within 15 iterations for a MIMO milling circuit. The research shows that objective functions can be constructed to promote either set point tracking or disturbance rejection of a controller. ITAE based objective functions are found to be well suited for the ore milling circuit given the large time constants. The method shows potential to optimally tune existing PI controllers of MIMO systems using robust stability margins to constrain the parameter search domain. Future work will evaluate an approach to reduce the impact on plant downtime during optimisation iterations and present comparisons against alternative methods such as reinforcement learning.

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.

Data availability

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

References

[1] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, Control Eng. Pract. 11 (7) (2003) 733–764.
[2] L. Desborough, R. Miller, Increasing customer value of industrial control performance monitoring - Honeywell's experience, in: AIChE Symposium Series, No. 326, New York; American Institute of Chemical Engineers; 1998, 2002, pp. 169–189.
[3] M. He, W. Cai, B. Wu, M. He, Simple decentralized PID controller design method based on dynamic relative interaction analysis, Ind. Eng. Chem. Res. 44 (2005) 8334–8344.
[4] J.G. Ziegler, N.B. Nichols, Optimum setting for automatic controllers, Trans. ASME 64 (1942) 759–768.
[5] G.H. Cohen, G.A. Coon, Theoretical consideration of retarded control, Trans. ASME 75 (1953) 827–834.
[6] C.E. Garcia, M. Morari, Internal model control. a unifying review and some new results, Ind. Eng. Chem. Process Des. Dev. 21 (2) (1982) 308–323.
[7] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, J. Process Control 13 (4) (2003) 291–309.
[8] K.J. Åström, T. Hägglund, Revisiting the Ziegler–Nichols step response method for PID control, J. Process Control 14 (6) (2004) 635–650.
[9] W.L. Luyben, Process Modeling, Simulation and Control for Chemical Engineers, second ed., McGraw-Hill Publishing Company, 1990, pp. 594–606.
[10] K.J. Åström, T. Hägglund, Automatic tuning of simple regulators with specifications on phase and amplitude margins, Automatica 20 (5) (1984) 645–651.
[11] C.C. Hang, A.P. Loh, V.U. Vasnani, Relay feedback auto-tuning of cascade controllers, IEEE Trans. Control Syst. Technol. 2 (1) (1994) 42–45.
[12] C.-C. Hang, Q. Wang, L.-S. Cao, Self-tuning smith predictors for processes with long dead time, Internat. J. Adapt. Control Signal Process. 9 (3) (1995) 255–270.
[13] Q.-G. Wang, C.-C. Hang, Q. Bi, Process frequency response estimation from relay feedback, Control Eng. Pract. 5 (9) (1997) 1293–1302.
[14] Q.-G. Wang, C.-C. Hang, B. Zou, Low-order modeling from relay feedback, Ind. Eng. Chem. Res. 36 (2) (1997) 375–381.
[15] Q.-G. Wang, B. Zou, T.-H. Lee, Q. Bi, Auto-tuning of multivariable PID controllers from decentralized relay feedback, Automatica 33 (3) (1997) 319–330.
[16] C. Hang, K. Astrom, Q. Wang, Relay feedback auto-tuning of process controllers - a tutorial review, J. Process Control 12 (1) (2002) 143–162.
[17] H.-P. Huang, J.-C. Jeng, K.-Y. Luo, Auto-tune system using single-run relay feedback test and model-based controller design, J. Process Control 15 (6) (2005) 713–727.
[18] J. Berner, K. Soltesz, T. Hägglund, K.J. Åström, An experimental comparison of PID autotuners, Control Eng. Pract. 73 (2018) 124–133.

[19] R. Nian, J. Liu, B. Huang, A review on reinforcement learning: Introduction and applications in industrial process control, Comput. Chem. Eng. 139 (2020) 106886.
[20] M.N. Howell, M.C. Best, On-line PID tuning for engine idle-speed control using continuous action reinforcement learning, Control Eng. Pract. 8 (2) (2000) 147–154.
[21] H. Boubertakh, M. Tadjine, P.-Y. Glorennec, S. Labiod, Tuning fuzzy PD and PI controllers using reinforcement learning, ISA Trans. 49 (4) (2010) 543–551.
[22] P. Kofinas, A.I. Dounis, Online tuning of a PID controller with a fuzzy reinforcement learning MAS for flow rate control of a desalination unit, Electronics 8 (2) (2019) 231.
[23] O. Dogru, K. Velswamy, F. Ibrahim, Y. Wu, A.S. Sundaramoorthy, B. Huang, S. Xu, M. Nixon, N. Bell, Reinforcement learning approach to autonomous PID tuning, Comput. Chem. Eng. 161 (2022) 107760.
[24] 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.
[25] J.D. Le Roux, I.K. Craig, D. Hulbert, A. Hinde, Analysis and validation of a run-of-mine ore grinding mill circuit model for process control, Miner. Eng. 43 (2013) 121–134.
[26] 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.
[27] D. Wei, I.K. Craig, Grinding mill circuits - a survey of control and economic concerns, Int. J. Miner. Process. 90 (1–4) (2009) 56–66.
[28] M. Bauer, I.K. Craig, Economic assessment of advanced process control - a survey and framework, J. Process Control 18 (1) (2008) 2–18.
[29] F. Berkenkamp, A. Krause, A.P. Schoellig, Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics, Mach. Learn. (2021) 1–35.
[30] M. Fiducioso, S. Curi, B. Schumacher, M. Gwerder, A. Krause, Safe contextual Bayesian optimization for sustainable room temperature PID control tuning, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 5850–5856.
[31] A. Lucchini, S. Formentin, M. Corno, D. Piga, S.M. Savaresi, Torque vectoring for high-performance electric vehicles: An efficient MPC calibration, IEEE Control Syst. Lett. 4 (3) (2020) 725–730.
[32] F. Sorourifar, G. Makrygirgos, A. Mesbah, J.A. Paulson, A data-driven automatic tuning method for MPC under uncertainty using constrained Bayesian optimization, IFAC-PapersOnLine 54 (3) (2021) 243–250.
[33] Q. Lu, L.D. González, R. Kumar, V.M. Zavala, Bayesian optimization with reference models: a case study in MPC for HVAC central plants, Comput. Chem. Eng. 154 (2021) 107491.
[34] J.A. Paulson, G. Makrygiorgos, A. Mesbah, Adversarially robust Bayesian optimization for efficient auto-tuning of generic control structures under uncertainty, AIChE J. 68 (6) (2022) e17591.
[35] E. Brochu, V. Cora, N. Freitas, A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, 2010.
[36] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, in: Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 2, 2012, pp. 2951–2959.
[37] R. Lam, M. Poloczek, P. Frazier, K.E. Willcox, Advances in Bayesian optimization with applications in aerospace engineering, in: 2018 AIAA Non-Deterministic Approaches Conference, 2018, p. 1656.
[38] R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, I. Guyon, Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020, in: NeurIPS 2020 Competition and Demonstration Track, PMLR, 2021, pp. 3–26.
[39] 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-PapersOnLine 55 (21) (2022) 126–131.
[40] S. Skogestad, I. Postlethwaite, Multivariable Feedback Control: Analysis and Design, Vol. 2, Citeseer, 2007.
[41] D.E. Seborg, T.F. Edgar, D.A. Mellichamp, F.J. Doyle, Process Dynamics and Control, John Wiley & Sons, 2011.
[42] J. Pongfai, X. Su, H. Zhang, W. Assawinchaichote, PID controller autotuning design by a deterministic Q-SLP algorithm, IEEE Access 8 (2020) 50010–50021.
[43] J.T. Wilson, F. Hutter, M.P. Deisenroth, Maximizing acquisition functions for Bayesian optimization, in: 32nd Conference on Neural Information Processing Systems, Vol. 20, No. 5, 2018, pp. 645–651.
[44] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2) (2012) 281–305.
[45] A.D. Bull, Convergence rates of efficient global optimization algorithms, J. Mach. Learn. Res. 12 (10) (2011) 2879–2904.
[46] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006, pp. 7–29.
[47] E.R. Ackermann, J.P. De Villiers, P. Cilliers, Nonlinear dynamic systems modeling using Gaussian processes: Predicting ionospheric total electron content over South Africa, J. Geophys. Res. Space Phys. 116 (A10) (2011) A10303.
[48] K. Ažman, J. Kocijan, Application of Gaussian processes for black-box modelling of biosystems, ISA Trans. 46 (4) (2007) 443–457.
[49] B. Liu, Q. Zhang, G.G. Gielen, A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems, IEEE Trans. Evol. Comput. 18 (2) (2013) 180–192.
[50] 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) (2015) 148–175.
[51] J. Mockus, On the Bayes methods for seeking the extremal point, IFAC Proc. Vol. 8 (1, Part 1) (1975) 428–431.
[52] M.T. Emmerich, K.C. Giannakoglou, B. Naujoks, Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels, IEEE Trans. Evol. Comput. 10 (4) (2006) 421–439.
[53] D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions, J. Global Optim. 13 (1998) 455–492.
[54] I. Craig, D. Hulbert, G. Metzner, S. Moult, Extended particle-size control of an industrial run-of-mine milling circuit, Powder Technol. 73 (3) (1992) 203–210.
[55] I.K. Craig, I.M. MacLeod, Specification framework for robust control of a run-of-mine ore milling circuit, Control Eng. Pract. 3 (5) (1995) 621–630.
[56] L.C. Coetzee, I.K. Craig, E.C. Kerrigan, Robust nonlinear model predictive control of a run-of-mine ore milling circuit, IEEE Trans. Control Syst. Technol. 18 (1) (2010) 222–229.
[57] E. Bristol, On a new measure of interaction for multivariable process control, IEEE Trans. Automat. Control 11 (1) (1966) 133–134.
[58] I.K. Craig, D.G. Hulbert, G. Metzner, S.P. Moult, Optimised multivariable control of an industrial run-of-mine milling circuit, J. South. Afr. Inst. Min. Metall. 92 (6) (1992) 169–176.
[59] J.D. Le Roux, I.K. Craig, Plant-wide control framework for a grinding mill circuit, Ind. Eng. Chem. Res. 58 (26) (2019) 11585–11600.
[60] MATLAB, Version 9.12.0 (R2022a), The MathWorks Inc., Natick, Massachusetts, 2022.
[61] O. Galán, G. Barton, J. Romagnoli, Robust control of a SAG mill, Powder Technol. 124 (3) (2002) 264–271.
[62] J. Karageorgos, P. Genovese, D. Baas, Current trends in SAG and AG mill operability and control, in: Proceedings of an International Conference on Autogenous and Semiautogenous Grinding Technology, Vol. 3, 2006, pp. 191–206.