Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-21 (2022) 126–131

On-line automatic controller tuning using Bayesian optimisation - a bulk tailings treatment plant case study

J. A. van Niekerk [*,**] **J. D. le Roux** [**] **I. K. Craig** [**,1]

[*] Zutari, Pretoria, South Africa.
[**] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa.

Abstract: The automatic tuning problem of multiple-input-multiple-output (MIMO) controllers is considered within the framework of Bayesian optimisation and applied in simulation to a bulk tailings treatment process. The aim is to develop a model free, on-line, automatic 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 the process has changed. Simulations indicate that the method is able to locate the optimal tuning parameters for the bulk tailings treatment process as compared to a de-coupled controller developed from a model of the process. The parameters were obtained from an objective function which was balanced and weighted according to the response required.

Keywords: Bayesian optimisation, Gaussian processes, acquisition function, auto tuning, bulk tailings treatment

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 (Qin and Badgwell, 2003), PID is by far the most common feedback controller. A survey of eleven thousand controllers in the continuous process industry indicated that 97% of those controllers implemented the PID algorithm (Desborough and Miller, 2002). Only a third of these controllers provided 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 requires the attention of domain experts. It is therefore evident that a need exists to obtain optimal tuning parameters for industrial controllers in an inexpensive manner.

Early auto-tuning methods such as the relay feedback method (Åström and Hägglund, 1984) were primarily intended to tune simple regulators of the PID type. Subsequently the relay feedback method has received much research and development attention resulting in the expansion of its application. In addition to the critical gain and critical period parameters, more information on process dynamics can be obtained from the same relay feedback test using new identification techniques (Hang et al., 2002). Machine learning has since expanded the possibilities of auto-tuning controllers by introducing self-learning techniques such as reinforcement learning (Nian et al., 2020).

Continuous action reinforcement learning automata (CARLA) were developed as one of the first reinforcement learning auto-tuning algorithms (Howell and Best, 2000). CARLA was implemented to auto-tune Ford Motors Zetec engines and showed a 60% improvement. By taking advantage of the on-line and model free learning properties of reinforcement learning, an auto-tuning PID controller was developed by Wang et al. (2007). The reinforcement Q-learning algorithm was used to auto-tune fuzzy PD and PI controllers of a simulated inverted pendulum and CE150 helicopter models (Boubertakh et al., 2010). A hybrid Zeigler-Nichols fuzzy reinforcement learning multi-agent system was used by Kofinas and Dounis (2019) to control the flow rate of a desalination unit. The gains of the controller were initialised using the Zeigler-Nichols method and then adapted on-line using reinforcement learning. Shipman and Coetzee (2019) applied reinforcement learning using deep neural networks to automatically tune a PI controller suitable for use over a wide range of plant models by changing the plant dynamics, disturbance and measurement noise during the training process.

Reinforcement learning agents adjust the tuning parameters of controllers in an adaptive way based on the best reward associated with the observed states of the environment, making the reinforcement learning agent and controller combination potentially well suited for complex non-linear systems. Reinforcement learning based auto-tuning, however, does not provide a single, optimal set of tuning parameters, since the parameters are continuously adapted to the changing states within the environment. The many training steps required to train reinforcement learning agents could be impractical in processes with significant time delays and long settling times as this will

[1] Corresponding author. E-mail: ian.craig@up.ac.za.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2022.09.255

lead to substantial production down-time while training is in progress.

An ideal auto-tuner operates on-line, is model free, controller agnostic, data efficient and globally optimal (Neumann-Brosig et al., 2020). Bayesian optimisation (BO) is proposed as a method to auto tune controllers due to its on-line sampling characteristics and capability to obtain the global minimum of an objective function in only a few steps (Bull, 2011).

BO has been demonstrated to optimise the tuning parameters of a quadrotor vehicle (Berkenkamp et al., 2021). Neumann-Brosig et al. (2020) used BO 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 hand tuning after only 10 experiments. Fiducioso et al. (2019) used safe contextual BO to optimise the PID parameters of a room temperature controller without human intervention. Lucchini et al. (2020) and Sorourifar et al. (2021) applied BO to tune MPCs for torque vectoring of high performance electrical vehicles and a continuously stirred tank reactor respectively, to notably improve performance.

This paper demonstrates a machine learning technique that can automatically tune existing process controllers on-line, without having to identify the process model. The output of the auto-tuner is an optimised set of controller tuning parameters that can be applied to the target controller on completion of the auto-tuning procedure. The auto-tuner must interact with the live process, sample proposed tuning parameters, evaluate the performance of the proposed parameters and repeat the process until the performance objective function has been minimised. Process sampling must be cost efficient and be conducted in as few as possible steps. In this paper, BO is used to tune decentralised PID control loops that control a MIMO bulk tailings treatment process in simulation. An appropriate objective function is derived that is minimised by searching for the optimal tuning parameters.

This 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. Background information on BO, Gaussian processes and acquisition functions is provided in Section 3. Section 4 describes the process to be controlled and the controller structure used. Minimisation of the objective function by means of BO is demonstrated by simulation and benchmarked against results from Rokebrand et al. (2021). The results are discussed in Section 5 and concluding remarks provided in Section 6.

2. PROBLEM STATEMENT

Consider a dynamic MIMO process of an industrial plant that is to be controlled by a feedback controller. 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 $\mathcal{A} \subseteq \mathbb{R}$. The performance of the tuning parameters can be quantified by evaluating each of the observed process variables in terms of time domain performance indices.

Where multiple performance indices are used to evaluate a controller, they must be balanced 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

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

where Q is the objective function, n is the total count of process variables, ω_i is the process variable performance weighting, p is the total count of performance indices selected per process variable, Q_j is the performance index and β_{ij} is a balancing factor to scale the contribution of each performance index.

The form of the performance indices as functions of the tuning parameters is unknown, but can be calculated from experiments conducted on the process. 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 an optimisation function to find the set of tuning parameters that minimises the objective function Q

$$\min_{\alpha \in \mathcal{A}} = Q(\boldsymbol{\alpha}) \tag{2}$$

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

3. BACKGROUND

The Bayesian approach to optimisation is to first specify prior knowledge about the objective function using a probabilistic surrogate model, and then to locate the global minimum of that model using an acquisition function (Wilson et al., 2018). The surrogate model is computationally cheaper to evaluate and optimise compared to the unknown objective function. In this work the surrogate is modelled as a Gaussian process (Rasmussen and Williams, 2006). Gaussian processes not only provide predictions of unsampled inputs, but also the confidence of those predictions that can be interpreted in a natural way (Ackermann et al., 2011). 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 (Ažman and Kocijan, 2007). The computational complexity of Gaussian processes increases cubically as the number of sampling points increase (Liu et al., 2013), but since it is an objective to limit the number of expensive experiments, this limitation is not a concern.

3.1 Gaussian Processes

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

$$Q(\boldsymbol{\alpha}) \sim \mathcal{GP}(m(\boldsymbol{\alpha}), k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')) \tag{3}$$

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. The unrealistic smoothness of the commonly used squared exponential function makes it impractical for optimisation problems. Snoek et al. (2012) propose the automatic relevance determination (ARD) Matérn parameter 5/2 kernel as the covariance function.

Gaussian processes learn the input-output relationships from a training dataset. For the problem statement defined in (1) and (2), 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

$$\widehat{Q} = Q(\boldsymbol{\alpha}) + \varepsilon \tag{4}$$

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

$$\varepsilon \sim \mathcal{N}(0, \sigma_n^2). \tag{5}$$

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

$$\begin{bmatrix} \widehat{\boldsymbol{Q}} \\ \boldsymbol{Q}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(A,A) + \sigma_n^2 I & K(A, A_*) \\ K(A_*, A) & K(A_*, A_*) \end{bmatrix}\right) \tag{6}$$

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.

$$\boldsymbol{Q}_*|A, \widehat{\boldsymbol{Q}}, A_* \sim \mathcal{N}(\bar{\boldsymbol{Q}}_*, \mathrm{cov}(\boldsymbol{Q}_*)), \text{where} \tag{7}$$

$$\bar{\boldsymbol{Q}}_* = \boldsymbol{k}_*^\top [K + \sigma_n^2 I]^{-1} \widehat{\boldsymbol{Q}}, \tag{8}$$

$$\mathrm{cov}(\boldsymbol{Q}_*)) = \boldsymbol{k}_{**} - \boldsymbol{k}_*^\top [K + \sigma_n^2 I]^{-1} \boldsymbol{k}_*. \tag{9}$$

$\bar{\boldsymbol{Q}}_*$ 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 BO, 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. Acquisition functions identify the next input location to be sampled by finding the point where the acquisition function $\mathcal{L}$ is maximised, with (Snoek et al., 2012)

$$\boldsymbol{\alpha}_* = \underset{\boldsymbol{\alpha} \in \mathcal{A}}{\mathrm{argmax}}\, \mathcal{L}(\boldsymbol{\alpha}|\mathcal{D}) \tag{10}$$

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 (Snoek et al., 2012). In this work EI (Mockus, 1975) is selected, as it has been shown to escape local optimums (Emmerich et al., 2006), is better behaved than probability of improvement, and does not require a tuning parameter such as the Gaussian process upper confidence bound (Snoek et al., 2012).

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

$$\mathrm{EI}(\boldsymbol{\alpha}) = \mathbb{E} \max[0, \widehat{Q}(\boldsymbol{\alpha}_{min}) - \widehat{Q}(\boldsymbol{\alpha})] \tag{11}$$

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 (Jones et al., 1998) as

$$\mathrm{EI}(\boldsymbol{\alpha}) = \begin{cases} \Psi(\boldsymbol{\alpha}) + \Phi(\boldsymbol{\alpha}), & \text{if } \sigma(\boldsymbol{\alpha}) > 1 \\ 0, & \text{if } \sigma(\boldsymbol{\alpha}) = 0 \end{cases} \tag{12}$$

where

$$\Psi(\boldsymbol{\alpha}) = (\widehat{Q}(\boldsymbol{\alpha}_{min}) - \bar{\boldsymbol{Q}}_*(\boldsymbol{\alpha}))\psi(Z), \tag{13}$$

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

$$Z = \frac{(\widehat{Q}(\boldsymbol{\alpha}_{min}) - \bar{\boldsymbol{Q}}_*(\boldsymbol{\alpha}))}{\sigma(\boldsymbol{\alpha})}. \tag{15}$$

$\sigma(\boldsymbol{\alpha})$ 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. Equation (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,..., **do**
2:	select new $\boldsymbol{\alpha}_*$ by maximizing acquisition function $\mathcal{L}$ $\boldsymbol{\alpha}_* = \mathrm{argmax}_{\boldsymbol{\alpha} \in \mathcal{A}}\, \mathcal{L}(\boldsymbol{\alpha}|\mathcal{D}_n)$
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**

BO is a cyclic process that progresses as presented in Algorithm 1.

4. SIMULATION

4.1 Plant

The controller selected to optimise by means of BO is the controller for a bulk tailings treatment (BTT) plant as described by Rokebrand et al. (2021). A brief introduction of the process is provided here.

Fig. 1 illustrates the BTT surge tank process flow. The surge tank is fed with chrome tailings from the tailings dam at a feedrate q_i and density ρ_i. The tailings are diluted with water at a flow rate of q_w and agitated in the surge tank to promote mixing. The tank volume is v. The control objective is to stabilise the chrome concentrator supply density which makes use of spiral concentrators to separate chrome grades. A stable density supply to the concentrator improves separation efficiencies. The tank output feedrate is q_o and density ρ_o. The tank output feedrate is held constant at 750 m^3/hr. Perfect mixing is assumed and therefore $\rho_o = \rho$, where ρ is the density measured in the surge tank. The chrome tailings density ρ_i is not constant and is the disturbance that must be

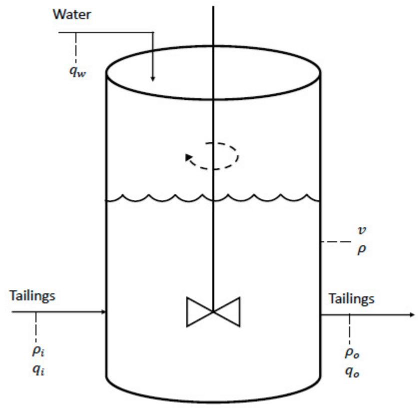

Fig. 1. Bulk tailings treatment surge tank (Rokebrand et al., 2021).

rejected by the process controller. The process variables to be controlled are the surge tank volume v and density ρ. The manipulated variables are the water flow rate q_w and tailings supply flow rate q_i.

The BTT transfer function matrix model in the form of

$$\boldsymbol{y} = \boldsymbol{G}_p(s)\boldsymbol{u} + \boldsymbol{G}_d(s)d \tag{16}$$

is given by

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \dfrac{1}{s} & \dfrac{1}{s} \\ \dfrac{0.01}{s+75} & \dfrac{-0.04}{s+75} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{60}{s+75} \end{bmatrix} d \tag{17}$$

which is linearised around an equilibrium operating point where $\boldsymbol{y} = [v, \rho]^\top$, $\boldsymbol{u} = [q_i, q_w]^\top$, and $d = \rho_i$.

4.2 Controller

The plant is controlled in closed-loop by controller C which is structured as a decentralised controller with PI controllers on the diagonal.

$$C = \begin{bmatrix} c_{11} & 0 \\ 0 & c_{22} \end{bmatrix} \tag{18}$$

The PI controllers in Laplace domain are of the form

$$c_{jj} = k_{pjj}(1 + \frac{1}{\tau_{ijj}s}), j = 1, 2 \tag{19}$$

where k_p is the proportional gain, and τ_i is the integral time. The tuning parameters to be optimised are k_{p11}, τ_{i11}, k_{p22} and τ_{i22}.

The Rokebrand et al. (2021) de-coupled PI controller against which the BO results are benchmarked is

$$\boldsymbol{C}_{Rokebrand} = \begin{bmatrix} \dfrac{84(s+50)}{s} & 0 \\ 0 & \dfrac{-1505.7(s+75)}{s} \end{bmatrix}. \tag{20}$$

4.3 Constraints

BO is a constrained regression process, and the constraints must be considered with care. For the purposes of this simulation the constraints are the ranges of the tuning parameters within which the BO algorithm must search for the optimal tuning parameters to minimize the objective function. The search space can be defined by either selecting the existing tuning parameters or identifying new ones by conducting step tests.

Step tests can be conducted to determine the magnitude and direction of the process gains as well as the time constants. Given the gains and time constants the candidate tuning parameters can be sought using any of the known PID tuning methods such as SIMC (Skogestad, 2003). The observations need not be very accurate since they will be used to determine the constraints around the candidate and not the optimal set of parameters.

To expand the search space around the identified tuning parameters, the gain constraints are conservatively selected as a factor of 2 in the direction of instability, and boldly selected as a factor of 0.2 in the opposite direction. Selection of the integral time constraints follows the inverse approach, i.e. a factor of 0.5 in the direction of instability and a factor of 5 in the opposite direction. It is possible that an optimum still exists beyond the search space but as that optimum is approached, the possibility of instability increases. For the objective functions selected, an unstable controller will not return a measurable value, so one needs to limit the number of unstable iterations to take advantage of the ability of the acquisition function to select the next sampling point.

For the purposes of the simulation, the search space is defined by selecting the tuning parameters from Rokebrand et al. (2021) and expanding the space around them as described. The constraints selected are

$$k_{p11} \in [16.8, 168] \tag{21}$$
$$\tau_{i11} \in [0.01, 0.1] \tag{22}$$
$$k_{p22} \in [-3010, -301] \tag{23}$$
$$\tau_{i22} \in [0.0067, 0.067] \tag{24}$$

A safe BO algorithm SafeOPT has been develop by Sui et al. (2015) and was further expanded on by Berkenkamp et al. (2021) to address multiple safety constraints. SafeOPT is suited for processes where exploration and exploitation by the BO algorithm could lead the equipment damage or pose a risk to personnel safety. SafeOPT comes at the expense of additional iterations required to expand the constraints within which the optimal parameters can be safely sought. The SafeOPT algorithm was not selected for the BTT process since unstable controller performance will neither impact the safety of equipment nor personnel.

4.4 Objective function

The MIMO objective function was obtained by first finding a convex objective function for the SISO controllers c_{jj} in (19), and then applying it to the MIMO controller of (18). Two objective functions were identified for discussion. The first objective function was constructed by only considering the settling time in both output channels with equal unitary weights. The first objective function in the form of (1) is

$$Q = T_{s1} + T_{s2} \tag{25}$$

where T_s is the settling time, i.e. the time taken for the error to stay within 2% of $|y_{final} - y_{initial}|$. The second objective function was constructed from a combination of settling time and peak usage

$$Q = \sum_{i=1}^{2} (0.6 \times T_{si} + (1.25 \times 10^{-4} \times u_{peaki})^2) \qquad (26)$$

where u_{peak} is the maximum controller output.

The frequency of the iterations is determined by the choice of objective function. Due to the objective functions selected, the tuning parameters cannot converge at a rate faster than the process time constants of 1/75 hours. Both objective functions rely on the closed-loop settling time, i.e. the time the error takes to reach and remain within the 2% tolerance region. The settling time must be measurable so that the objective function can return a valid scalar quantity to the update the posterior distribution.

5. RESULTS

Table 1 shows the results of the BO simulation using the objective function from (25), iterations 31 through to 40. Column Q represents the objective function value for each set of tuning parameters applied. The global minimum of the objective function is found by iteration 38 with the optimal tuning parameters as shown in the highlighted row. Note that the iterations do not stop once the global minimum is located but continue until a predetermined number of iterations have been reached or the controller performance criteria have been met.

During simulation, the step test response was evaluated over a period of 6 minutes. Therefore in practice it would take a minimum of 8 hours to complete 40 iterations (as suggested in Table 1), with each iteration consisting of two step tests.

Table 1. Results of Bayesian optimisation simulations using the objective function from (25), iterations 31 through 40

Iter	Q	k_{p11}	τ_{i11}	k_{p22}	τ_{i22}
31	0.1953	163	0.0106	-2998	0.0425
32	0.2233	167	0.0128	-564	0.0164
33	0.1501	112	0.0131	-1763	0.0174
34	0.1496	132	0.0132	-2973	0.0298
35	0.2114	128	0.0991	-2996	0.0261
36	0.1718	89	0.0116	-1403	0.0071
37	0.2094	98	0.0148	-2960	0.0358
38	**0.0550**	**167**	**0.0116**	**-2444**	**0.0116**
39	0.2393	81	0.0998	-1085	0.0076
40	0.2612	168	0.0127	-2992	0.0570

Table 2 shows the results of the BO simulation using the objective function from (26), iterations 14 through to 23. The objective function global minimum is found by iteration 21. The increased rate of convergence was made possible by the use of balancing factors, β_{ij} of (1), to shape the objective function surface. In practice it would take a minimum of 4,6 hours to complete 23 iterations (as suggested in Table 2).

To evaluate the performance of the optimal tuning parameters, the ability of the controller to reject disturbances are benchmarked against results from Rokebrand et al. (2021).

Fig. 2 shows the disturbance rejection response of the controllers tuned using BO with the objective functions from (25) and (26) compared against the Rokebrand et al.

Table 2. Results of Bayesian optimisation simulations using the objective function from (26), iterations 14 through 23

Iter	Q	k_{p11}	τ_{i11}	k_{p22}	τ_{i22}
14	0.0974	160	0.0235	-1522	0.0124
15	0.1109	167	0.0256	-1502	0.0151
16	0.1016	167	0.0117	-1651	0.0126
17	0.1146	167	0.0311	-1521	0.0098
18	0.0983	165	0.0110	-1352	0.0117
19	0.0988	141	0.0103	-1488	0.0107
20	0.1068	144	0.0101	-1641	0.0126
21	**0.0889**	**155**	**0.0153**	**-1424**	**0.0115**
22	0.0915	144	0.0152	-1321	0.0120
23	0.1267	119	0.0122	-836	0.0116

(2021) de-coupled PI controller for a step disturbance of 0.1 t/m^3 in the chrome tailings density ρ_i. The responses in Fig. 2 and the Root Mean Squared Error (RMSE) calculated in Table 3 shows that the BO controller with objective function (25) suppresses both the level and the density disturbance significantly better than the de-coupled controller. Objective function (26) suppresses the level disturbance better than the de-coupled controller, but only shows a marginal improvement on the density disturbance rejection. The peak usage performance index in objective function (26) penalises actuator usage, hence the slower suppression of the density disturbance.

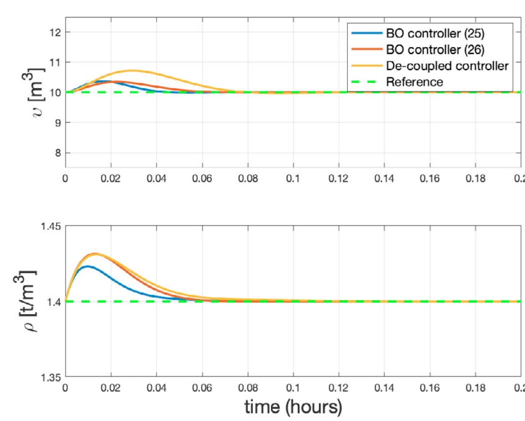

Fig. 2. Disturbance rejection response comparing the controller tuned using Bayesian optimisation with the objective functions from (25) and (26) against the Rokebrand et al. (2021) de-coupled PI controller.

Table 3. RMSE comparison of controller disturbance rejection

Controller	Volume RMSE	Density RMSE
De-coupled controller	0.2782	0.0105
BO controller (25)	0.1072	0.0066
BO controller (26)	0.1247	0.0101

6. CONCLUSION

BO is a data efficient, model free, on-line tuning method that can locate optimal tuning parameters within 21 iterations for a MIMO BTT plant. The disturbance response compared against the Rokebrand et al. (2021) de-coupled PI controller is favourable. The research shows that objective functions can be constructed, balanced and weighted according to the response required for the particular process. The method shows potential to optimally tune PI controllers for MIMO systems. Future work will evaluate the method for larger systems and will also aim to minimize the number of iterations of the BO routine to reduce the impact on plant performance.

REFERENCES

Ackermann, E.R., De Villiers, J.P., and Cilliers, P. (2011). Nonlinear dynamic systems modeling using Gaussian processes: Predicting ionospheric total electron content over South Africa. *Journal of Geophysical Research: Space Physics*, 116(A10), A10303.

Åström, K.J. and Hägglund, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20(5), 645–651.

Ažman, K. and Kocijan, J. (2007). Application of Gaussian processes for black-box modelling of biosystems. *ISA Transactions*, 46(4), 443–457.

Berkenkamp, F., Krause, A., and Schoellig, A.P. (2021). Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 1–35.

Boubertakh, H., Tadjine, M., Glorennec, P.Y., and Labiod, S. (2010). Tuning fuzzy PD and PI controllers using reinforcement learning. *ISA Transactions*, 49(4), 543–551.

Bull, A.D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10), 2879–2904.

Desborough, L. and Miller, R. (2002). Increasing customer value of industrial control performance monitoring - Honeywell's experience. In *AIChE Symposium Series*, 326, 169–189. New York; American Institute of Chemical Engineers; 1998.

Emmerich, M.T., Giannakoglou, K.C., and Naujoks, B. (2006). Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4), 421–439.

Fiducioso, M., Curi, S., Schumacher, B., Gwerder, M., and Krause, A. (2019). Safe contextual Bayesian optimization for sustainable room temperature PID control tuning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 5850–5856.

Hang, C., Astrom, K., and Wang, Q. (2002). Relay feedback auto-tuning of process controllers—a tutorial review. *Journal of Process Control*, 12(1), 143–162.

Howell, M.N. and Best, M.C. (2000). On-line PID tuning for engine idle-speed control using continuous action reinforcement learning. *Control Engineering Practice*, 8(2), 147–154.

Jones, D.R., Schonlau, M., and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, 455–492.

Kofinas, P. and Dounis, A.I. (2019). Online tuning of a PID controller with a fuzzy reinforcement learning MAS for flow rate control of a desalination unit. *Electronics*, 8(2), 231.

Liu, B., Zhang, Q., and Gielen, G.G. (2013). A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Transactions on Evolutionary Computation*, 18(2), 180–192.

Lucchini, A., Formentin, S., Corno, M., Piga, D., and Savaresi, S.M. (2020). Torque vectoring for high-performance electric vehicles: An efficient MPC calibration. *IEEE Control Systems Letters*, 4(3), 725–730.

Mockus, J. (1975). On the Bayes methods for seeking the extremal point. *IFAC Proceedings Volumes*, 8(1, Part 1), 428–431.

Neumann-Brosig, M., Marco, A., Schwarzmann, D., and Trimpe, S. (2020). Data-efficient autotuning with Bayesian optimization: an industrial control study. *IEEE Transactions on Control Systems Technology*, 28(3), 730–740.

Nian, R., Liu, J., and Huang, B. (2020). A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139, 106886.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.

Rasmussen, C.E. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rokebrand, L.L., Burchell, J.J., Olivier, L.E., and Craig, I.K. (2021). Competing advanced process control via an industrial automation cloud platform. *arXiv preprint arXiv:2011.13184*.

Shipman, W.J. and Coetzee, L.C. (2019). Reinforcement learning and deep neural networks for PI controller tuning. *IFAC-PapersOnLine*, 52(14), 111–116.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13(4), 291–309.

Snoek, J., Larochelle, H., and Adams, R.P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 2*, 2951–2959.

Sorourifar, F., Makrygirgos, G., Mesbah, A., and Paulson, J.A. (2021). A data-driven automatic tuning method for MPC under uncertainty using constrained Bayesian optimization. *IFAC-PapersOnLine*, 54(3), 243–250.

Sui, Y., Gotovos, A., Burdick, J.W., and Krause, A. (2015). Safe exploration for optimization within Gaussian processes. *Proceeding of the International Conference on Machine Learning*, 997–1005.

Wang, X., Cheng, Y., and Sun, W. (2007). A proposal of adaptive PID controller based on reinforcement learning. *Journal of China University of Mining & Technology*, 17(1), 40–44.

Wilson, J.T., Hutter, F., and Deisenroth, M.P. (2018). Maximizing acquisition functions for Bayesian optimization. *32nd Conference on Neural Information Processing Systems*, 20(5), 645–651.