ON-LINE AUTO-TUNING OF MULTIVARIABLE INDUSTRIAL PROCESSES USING BAYESIAN OPTIMISATION

by

Jonathan Anson van Niekerk

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

June 2023

SUMMARY

ON-LINE AUTO-TUNING OF MULTIVARIABLE INDUSTRIAL PROCESSES USING BAYESIAN OPTIMISATION

by

Jonathan Anson van Niekerk

Supervisor:	Prof. I. K. Craig
Co-supervisor:	Prof. J. D. le Roux
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Electronic Engineering)
Keywords:	acquisition function, auto-tuning, Bayesian optimisation, bulk tailings treatment, Gaussian processes, ore milling

Research shows that a significant number of industrial process controllers are poorly tuned. These poorly tuned controllers persist not because there is insufficient literature on tuning methods but rather because the research is not applied in practice due to the expense of subject matter experts and sub-optimal process performance experienced during model identification experiments. Evidently there is a need for auto-tuning of industrial processes controllers to optimise performance that does not demand the attention of subject matter experts.

Auto-tuning has been considered as early as 1984 by Åström and T. Hägglund when they presented the relay feedback method. With the advancement of computer processing capabilities alternative approaches such as machine learning including reinforcement learning and Bayesian optimisation have been introduced. The on-line, model free, controller agnostic, data efficient and globally optimal characteristics of Bayesian optimisation makes it an ideal candidate for auto-tuning.

The supporting theory of Bayesian optimisation is presented and the selection of Gaussian processes, Matérn parameter 5/2 kernel and expected improvement as surrogate model, covariance function and

acquisition function are motivated.

Auto-tuning of multi-input multi-output (MIMO) controllers of a bulk tailing treatment (BTT) surge tank is presented. Two controllers are selected for optimisation. The first controller is a decentralised proportional-integral (PI) controller that controls a plant simulated using a linear model. The second controller is a multivariable inverse PI controller that controls a plant simulated using a non-linear process model. Objective functions are designed to promote set point tracking and disturbance rejection. The search domain constraints are determined by intuitively expanding the search domain around the tuning parameters of the reference controller. Results show that Bayesian optimisation is successful in improving the performance of the set point tracking and disturbance rejection controllers for the surge tank process.

Given the success of Bayesian optimisation on the surge tank controller, a more demanding control application in the form of an ore milling circuit controller is presented. A decentralised PI controller and μ-controller are selected for optimisation on the milling process. Objective functions are designed to promote set point tracking and disturbance rejection. A robust stability analysis is conducted to determine the constraints of the search domain. Results show that Bayesian optimisation is successful in improving the performance of the set point tracking and disturbance rejection controllers for the ore milling circuit.

Although the Bayesian optimisation framework can be applied to feedback controllers with real tuning parameters in general, this research only focusses on diagonal PI controllers and multivariable inverse controllers.

It is concluded that Bayesian optimisation can successfully auto-tune controllers for both the surge tank and ore milling processes for improved set point tracking and disturbance rejection performance. Objective functions can be designed to promote specific controller performance requirements. The search domain can be expanded to improve the probability of including the optimal tuning parameters whilst remaining within the stability margins of the controller.

LIST OF ABBREVIATIONS

ADRC	active disturbance rejection controller
ARD	automatic relevance determination
BLT	biggest log-modulus tuning
BTT	bulk tailings treatment
EI	expected improvement
MIMO	multi-input multi-output
PID	proportional integral derivative
RGA	relative gain array
RMSE	root mean square error
ROM	run of mine
SAG	semi-autogenous
SISO	single-input single-output

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

Process controllers and especially proportional-integral-derivate (PID) controllers are abundant in the industry due to their simplicity (Desborough and Miller, 2002). PID controllers are versatile and can be implemented as single-input single-output (SISO) controllers, be part of a de-centralised multi-input multi-output (MIMO) controllers, multivariable controllers, or perform the base level control of advanced process controllers. Many classical tuning methods exists for tuning PID controllers including among others Ziegler–Nichols (1942), Cohen–Coon (1953), IMC (Garcia and Morari, 1982), SIMC (Skogestad, 2003), and AMIGO (Åström and Hägglund, 2004).

The common objective of the mentioned tuning methods is to determine the optimal parameters for a PID controller. These methods aim to achieve desired control performance by adjusting the controller settings based on system dynamics and response characteristics. The Ziegler–Nichols method involves the identification two critical parameters, the ultimate gain and period, by inducing sustained oscillations in the system. Based on these values, tuning parameters are determined for proportional, proportional-integral, and proportional-integral-derivative controllers. The Cohen–Coon method involves analysing the process response of a step test to determine the ultimate gain and ultimate time constant. These values are then used to calculate controller parameters. The IMC method focuses on achieving desired closed-loop performance by incorporating the process dynamics into the controller design. It provides a systematic and model-based approach to controller tuning, offering improved control performance for a wide range of systems. The SIMC (Simple IMC) tuning method is a simplified version of the IMC approach used for tuning PID controllers. SIMC aims to provide a quick and practical tuning solution by approximating the dynamic behaviour of the process with a

first-order-plus-time-delay model. The AMIGO method involves determining the optimal controller settings based on the process response to a step test. AMIGO utilises a combination of analytical equations and iterative optimisation to find the appropriate values for the controller parameters. The method aims to achieve desired closed-loop performance by minimizing integral error and ensuring stability.

Despite the published tuning methods at the disposal of industry, poorly tuned controllers remain prevalent. The research of Desborough and Miller (2002) shows that only a third of PID controllers provide an acceptable level of performance. It is therefore reasonable to assume that industrial PID controllers are often only tuned when installed and then just left as is. This is partly due to the fact that the process of tuning controllers require the consultation of domain experts and sub-optimal process performance.

It is within this contexts that the opportunity exists to address the problem of poorly tuned controllers by auto-tuning industrial process controllers to meet specified performance criteria without the attention of domain experts. Neumann-Brosig, Marco, Schwarzmann and Trimpe (2020) proposes an on-line automatic controller tuning method using Bayesian optimisation. The approach combines Gaussian process models with elementary process knowledge in the form of an active disturbance rejection control (ADRC) structure, which incorporates a dynamic model and an observer to capture unmodeled dynamics and disturbances. The key parameters of the ADRC are learned from experimental data using Bayesian optimisation. This combined approach strikes a balance between physical knowledge and data-driven learning, making it flexible and effective for a wide range of control problems. This framework established by Neumann-Brosig et al. is repurposed to the application of auto-tuning multivariable industrial process controllers.

Auto-tuning of industrial process controllers offers several benefits but also limitations that are worth considering.

The benefits include:

- Improved performance: Auto-tuning algorithms can optimise controller parameters, leading to improved control performance, faster response times, reduced overshoot, and better disturbance rejection (Åström and Hägglund, 1984; Kofinas and Dounis, 2019).

- Time and cost savings: Auto-tuning eliminates the need for subject matter experts to conduct system identification experiments or the time and labour costs associated with trial-and-error methods (Desborough and Miller, 2002).

Limitations include:

- Model dependence: The approach followed in this research requires a process model to be determine the constraints of the search domain containing the optimal tuning parameters. Such models may not always be available or may be difficult to develop.
- Sub-optimal production: Auto-tuning algorithms introduce perturbations to observe the response of the process and evaluate the controller behaviour using different tuning parameters. Such perturbations will result in process variations and may result in sub-optimal production for the duration of the auto-tuning process.
- Personnel and equipment safety: Allowing auto-tuning algorithms to adjust controller tuning parameters may lead to unintentional controller instability that could result in damage to equipment or personnel injury. This may limit the application of auto-tuning algorithms to process that are inherently safe to perturb unless the tuning parameter search space is constrained conservatively so as not to introduce unstable perturbations.

1.1.2 Research gap

The implementation of Bayesian optimisation to auto-tune controllers has gained significant research attention in recent years and is made possible by the advancements in computer processing capabilities. Bayesian optimisation is used by Berkenkamp, Krause and Schoellig (2021) to auto-tune an aerial quad-rotor vehicle, Neumann-Brosig et al. (2020) to auto-tune an active disturbance rejection controller (ADRC), Fiducioso, Curi, Schumacher, Gwerder and Krause (2019) to auto-tune a room temperature controller, Lucchini, Formentin, Corno, Piga and Savaresi (2020) for auto-tuning of torque vectoring of electrical vehicles and Sorourifar, Makrygirgos, Mesbah and Paulson (2021) to auto-tune a continuously stirred tank reactor to improve performance.

Research addressing the application of Bayesian optimisation to auto-tune MIMO controllers of industrial processes could not be found.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

Bayesian optimisation requires a surrogate model, covariance function, acquisition function, fit for purpose objective function to minimise and a constrained, multi-dimensional search domain from which to selected candidate sampling points. With these Bayesian optimisation prerequisites in mind, the research objectives are to:

- Motivate the selection of a surrogate model, covariance function and an acquisition function based on prior knowledge of the objective function.
- Formulate a methodology to design objective functions to capture controller performance requirements. The objective function must be designed in such a way that when minimised by Bayesian optimisation the controller will be optimised to meet the desired performance requirements.
- Motivate an approach to defining the constraints of the search domain. The search domain must be defined to allow the inclusion of sampling points to locate the global minimum of the objective function while minimising the probability of including sampling points that cause controller instability and potential equipment damage.
- Establish a simulation environment to auto-tune industrial process controllers to meet the specified performance demands.
- Benchmark the optimised controllers against the reference controllers to evaluate the performance improvement.

Research questions include:

- Which surrogate model, covariance function and acquisition function is most suitable to model the unknown objective function and find the global minimum?
- Can an objective function be designed to capture specific performance criteria, so that when minimised the optimised controller will demonstrate improvement of the criteria specified?
- What measurable performance criteria can be included in the design of the objective function to improve the set point tracking or disturbance rejection of the controller?”
- Is there a systematic approach to determine the search domain constraints that will include the global minimum of the objective function while minimising the probability of unstable iterations?

- Can the Bayesian optimisation approach to auto-tuning captured in literature be extended to MIMO industrial process controllers?
- How does the performance of auto-tuned controllers compare against controllers tuned with classical tuning methods?
- Does the Bayesian optimisation approach to auto-tuning industrial process controllers show potential for practical implementation?

1.3 APPROACH

The approach followed is:

- Research and present the recent trends in the field of autonomous controller tuning including the relay feedback method, reinforcement learning and Bayesian optimisation.
- Study and present the theory of Bayesian optimisation.
- Motivate the choice of surrogate model, covariance function and acquisition function.
- Select the bulk tailings treatment (BTT) surge tank multivariable - and decentralised proportional-integral (PI) controllers (Rokebrand, Burchell, Olivier and Craig, 2021) as references to auto-tune for optimal set point tracking and disturbance rejection.
- Simulate the Bayesian optimisation of the reference controller on both linear and non-linear BTT process models.
- Select the ore milling circuit decentralised PI controller and μ-controller (Craig and MacLeod, 1995; Coetzee, Craig and Kerrigan, 2010; Le Roux, Craig, Hulbert and Hinde, 2013) as reference controllers to present a greater auto-tuning challenge due to the increased dimensionality and stronger interactions between plant input and output variables.
- Simulate the Bayesian optimisation of the reference controller on the non-linear ore milling circuit.

1.4 RESEARCH GOALS

The research goals include:

- Present the approach to design objective functions.
- Present the approach to constrain the search domain.

- Demonstrate improved performance of the auto-tuned BTT controller.
- Demonstrate improved performance of the auto-tuned ore milling circuit controller.

1.5 RESEARCH CONTRIBUTION

Contribution to the existing body of research is made by:

- Applying Bayesian optimisation to automatically tune MIMO industrial process controllers.
- Developing a pragmatic approach to designing fit for purpose objective functions.
- Applying robust stability analyses to controllers with uncertain parameters to determine the constraints of the search domain.

1.6 RESEARCH OUTPUTS

The following articles were produced from this work:

- J.A. van Niekerk, J.D. le Roux, and I. K. Craig. "On-line automatic controller tuning using Bayesian optimisation - a bulk tailings treatment plant case study." IFAC-PapersOnLine 55, no. 21 (2022): 126-131.
- J.A. van Niekerk, J.D. le Roux, and I. K. Craig. "On-line automatic controller tuning for a multivariable plant using Bayesian optimisation - an ore milling case study", submitted to: Journal of Process Control, November 2022.

1.7 OVERVIEW OF STUDY

Chapter 2 presents the review of relevant literature and makes reference to the current state of controller tuning, classical tuning methods and advancements made in the field of autonomous controller tuning. Chapter 3 presents the problem statement of auto-tuning MIMO process controllers, provides background to Bayesian optimisation and describes the approach to autonomous controller tuning. Chapter 4 describes the process to design a fit for purpose objective function and implements the Bayesian approach and optimises BTT surge tank controllers. Chapter 5 applies a robust stability analysis to determine the search domain constraints and implements Bayesian optimisation tuning on a more challenging ore milling circuit controller. Chapter 6 provides the concluding remarks.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OVERVIEW

This chapter describes the current state of under performing process controllers in the industry and motivates the case for automated controller tuning. Research of established auto-tuning methods is referenced as well as the more recent machine learning based auto-tuning techniques enabled by the advancements made in computer processing capability. Bayesian optimisation is motivated as a candidate for the preferred auto-tuner and related research is presented.

2.2 A LITERATURE STUDY OF CONTROLLER AUTO-TUNING

Process controllers and especially 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 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 (Desborough and Miller, 2002). PID controllers are also implemented extensively as part of decentralised controllers for MIMO processes (He, Cai, Wu and He, 2005).

Numerous PID tuning methods have been researched and published. Better known methods include Ziegler-Nichols (1942), Cohen-Coon (1953), IMC (Garcia and Morari, 1982), SIMC (Skogestad, 2003), and AMIGO (Åström and Hägglund, 2004). Luyben (1990) 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. The BLT method relies on the Ziegler-Nichols method to determine initial tuning parameters which are then detuned to achieve system stability.

In spite of the abundance of PID tuning methods available to the industry, Desborough and Miller (2002) 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 sub-optimal process performance. 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. An optimally tuned controller is a general term that depends on the objective of the process and could for example be the quality of the product at the expense of throughput or the setpoint tracking ability of controllers on the lower level of the control hierarchy.

Auto-tuning is not a novel concept and has enjoyed much research since the relay feedback method of Åström and Hägglund (1984). 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. Hang, Loh and Vasnani (1994) extends the relay feedback auto-tuning technique to auto-tune cascade controllers. For processes with long dead-times, Hang, Wang and Cao (1995) shows that Smith predictor controllers can be auto-tuned by combining the relay feedback method to identify the process model and controller design. Wang, Hang and Bi (1997) shows that multiple points on the process frequency response can be identified using a single relay test and application of the fast Fourier transform. For processes characterised by the first-order-plus-time-delay model, Wang, Hang and Zou (1997) uses a biased relay feedback test to accurately identify the critical points and parameters of the model. Wang, Zou, Lee and Bi (1997) presents a method for auto-tuning fully cross-coupled multivariable PID controllers based on relay feedback for processes with significant interaction. Hang, Åström and Wang (2002) introduces a further modification of the relay feedback method to identify multiple process frequency response points by superimposing a parasitic relay to the standard relay. In addition to the critical gain and critical period more information on process dynamics can be obtained from the same relay feedback test using new identification techniques (Hang et al., 2002). Huang, Jeng and Luo (2005) applies a single run of the standard relay feedback experiment to identify first- or second-order-plus-time-delay process models. In light of the success of the relay feedback research, auto-tuners based on the method have been commercialised. The performance of two such auto-tuners is discussed in Berner, Soltesz, Hägglund and Åström (2018).

Advancements made in computer processing capabilities and machine learning have provided an alternative approach to auto-tuning controllers by introducing self-learning techniques such as reinforcement learning (Nian, Liu and Huang, 2020). Reinforcement learning has been used in auto-tuning applications since the turn of the century.

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 the Ford Motors Zetec engines and showed a 60% improvement over traditional methods. By taking advantage of the on-line and model free learning properties of reinforcement learning, an auto-tuning PID controller was developed by Wang, Cheng and Sun (2007). The reinforcement Q-learning algorithm was used to auto-tune fuzzy PD and PI controllers of a simulated inverted pendulum model and a CE150 helicopter model (Boubertakh, Tadjine, Glorennec and Labiod, 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 tuning 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. Lawrence, Stewart, Loewen, Forbes, Backstrom and Gopaluni (2020) expressed a PID controller as a shallow neural network in the actor of the reinforcement learning framework. The PID gains are the weights of the actor network and are optimised by maximizing the reward function.

Dogru, Velswamy, Ibrahim, Wu, Sundaramoorthy, Huang, Xu, Nixon and Bell (2022) applies reinforcement learning to tune PI controllers of a pilot scale, non-linear tank system. The approach requires step-response models for the required operating points on the non-linear system to obtain reference PI tuning parameters using off-line tuning rules. The parameters are constrained for safe exploration and used as a baseline for the reinforcement learning agent. The agent is trained using the reference PI parameters to maximise the reward observed from the off-line step test simulations. Once the desired off-line performance criteria is achieved, the agent is deployed on the on-line system for further training in real time while exposed to actual process dynamics. Once training is complete the agent is capable of providing a suitable set of PI parameters for any set point given by the user.

Neumann-Brosig et al. (2020) considers the ideal auto-tuner to be on-line, model free, controller

agnostic, data efficient and globally optimal (minimises or maximises the objective function depending on the objective). 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 sub-optimal process performance and production loss during performance evaluation.

This research investigates Bayesian optimisation as a candidate for ideal auto-tuning of industrial processes. Bayesian optimisation is used to optimally tune a decentralised PI controller, multivariable controller and μ-controller applied to industrial processes in simulation. 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. Although the relay method presented by Wang, Zou, Lee and Bi (1997) could be applied, the use of Bayesian optimisation is researched as an alternative approach. 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 quad-rotor vehicle (Berkenkamp et al., 2021). Neumann-Brosig et al. (2020) 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. Lucchini et al. (2020) and Sorourifar et al. (2021) 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. Fiducioso et al. (2019) used safe contextual Bayesian optimisation to optimise the PID parameters of a room temperature controller without human intervention. The safe contextual GP-LCB algorithm developed and implemented by Fiducioso et al. (2019) starts searching for optimal parameters withing a safe set. The safe set is a small set around initial parameters that is considered safe. The safe set expands as the algorithm reduces uncertainty by querying boundary points and includes points with high performance. As a comparison, this research applies robust stability to determine the safe constraints of the search domain.

Bayesian optimisation is well suited for the optimisation of unknown (i.e. black box) objective functions that are expensive to evaluate (Brochu, Cora and Freitas, 2010). The expense can be expressed in any sense including computational effort, production down-time, cost of expertise or

capital cost of evaluation. Snoek, Larochelle and Adams (2012) shows that Bayesian optimisation can reach or surpass human expert-level tuning of hyper-parameters for machine learning algorithms. Lam, Poloczek, Frazier and Willcox (2018) 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 hyper-parameters (Turner, Eriksson, McCourt, Kiili, Laaksonen, Xu and Guyon, 2021).

2.3 CHAPTER SUMMARY

The literature study concludes that, despite the abundance of PID tuning methods available to the process industry, a need exists to automatically tune controllers for optimal performance in a cost-effective manner. Various relay feedback and machine learning based methods are considered and ultimately Bayesian optimisation is selected as the research candidate because it is considered to have the characteristics of an ideal auto-tuner, has shown to optimise controllers of various applications and is well suited for expensive to evaluate unknown objective functions.

CHAPTER 3 THE BAYESIAN APPROACH TO AUTO-TUNING CONTROLLERS

3.1 CHAPTER OVERVIEW

This chapter describes the framework within which Bayesian optimisation can be used to optimally tune a feedback controller for a dynamic MIMO plant. The feedback controllers included in this research are decentralised PI controllers and multivariable PI controllers. The optimal tuning parameters are found by minimising an unknown objective function that represents the performance of the controller. An overview of Bayesian optimisation is provided, describing how the unknown objective function is approximated by a surrogate model. The global minimum of the surrogate model is then found by an acquisition function. The choice of Gaussian processes to model the surrogate and expected improvement as acquisition function is motivated.

3.2 PROBLEM STATEMENT

Consider a dynamic MIMO process of an industrial plant represented by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)) \tag{3.1a}$$

$$y(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{d}(t)) \tag{3.1b}$$

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

Assume that the plant of (3.1) is controlled by a controller in a unity feedback configuration as shown in Fig. 3.1. The controller can be represented by

$$\boldsymbol{u}(t) = \boldsymbol{K}(\boldsymbol{e}(t), \alpha) \tag{3.2}$$

were $\boldsymbol{K}$ represents the controllers, and $\boldsymbol{e}(t)$ is the control error vector or the difference between the set points $\boldsymbol{r}(t)$ and the observed process variables, and α represents the controller tuning parameters.

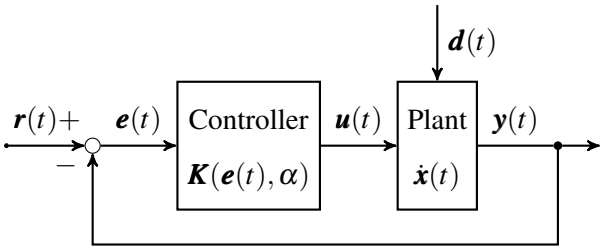

Figure 3.1. Feedback controller.

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 $\mathcal{A} \subseteq \mathbb{R}^d$, where d represents the dimensions of the domain. The performance of the tuning parameters are quantified in this work by evaluating each of the observed controlled variables or manipulated variables in terms of time domain performance indices. These performance indices for controller evaluation could typically include the integral of the squared error (ISE), integral of the absolute error (IAE), integral of the time multiplied squared error (ITSE), integral of the time multiplied absolute error (ITAE), rise time, settling time, overshoot, total variation, decay ratio, etc. (Skogestad and Postlethwaite, 2007; Seborg, Edgar, Mellichamp and Doyle, 2011; Pongfai, Su, Zhang and Assawinchaichote, 2020). These indices are dependent on the inputs used with step changes being the most typical input variation. Definitions of selected performance indices are provided in Table 3.1.

Where multiple performance indices are used to evaluate a controller, they must be scaled according to the required response. The performance associated with each controlled 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

$$Q = \sum_{i=1}^{n} \omega_i \left(\sum_{j=1}^{p} \beta_{ij} q_j(\boldsymbol{\alpha}) \right), \tag{3.3}$$

where Q is the objective function, n is the total number of controlled variables, ω_i is the controlled variable performance weighting, p is the total number of performance indices selected per controlled

Table 3.1. Definitions of selected performance indices.

Performance index	Definition
Rise time	Time it takes for the response to first reach 90% of its final value, which is usually required to be small (Skogestad and Postlethwaite, 2007).
Settling time	Settling time measures the time it takes for the error to stay below $\varepsilon\%$, where $\varepsilon = 2$ for this study with $\lvert y(t) - y_{final}\rvert \leq \varepsilon\% \times \lvert y_{final} - y_{initial}\rvert$ (Seborg et al., 2011).
Transient time	Transient time measures how quickly the transient dynamics die out i.e. for the error to stay below $\varepsilon\%$ of e_{max}, with $\lvert y(t) - y_{final}\rvert \leq \varepsilon\% \times e_{max}$ where e_{max} is the maximum error of $\lvert y(t) - y_{final}\rvert$ (MATLAB, 2022). Transient time is suitable to measure the duration of disturbances on non-stepped controlled variables.
Overshoot	The peak value divided by the final value, which should typically be 1.2 (20%) or less (Skogestad and Postlethwaite, 2007).
Peak	Peak value of $\lvert y(t) - y_{init}\rvert$ (MATLAB, 2022).
Total variation	The total up and down movement of the signal (input or output) which should be as small as possible (Seborg et al., 2011).
Decay ratio	The ratio of the second and first peaks, which should typically 0.3 or less. Interestingly the Cohen and Coon (1953) tuning method aims to achieve a decay ratio 0.25, referred to as quarter amplitude damping.
IAE	$IAE = \int_0^{T_{eval}} \lvert y(t) - y_{final}\rvert dt$ (Seborg et al., 2011), where T_{eval} is the evaluation period, which in the case of auto-tuning needs to be as short as possible to reduce the total evaluation period of all the optimisation iterations.
ISE	$ISE = \int_0^{T_{eval}} (y(t) - y_{final})^2 dt$ (Seborg et al., 2011)
ITAE	$ITAE = \int_0^{T_{eval}} t\lvert y(t) - y_{final}\rvert dt$ (Seborg et al., 2011)

variable, q_j is the performance index and β_{ij} is a scaling 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 (3.1). Candidate tuning parameters are identified and selected for each experiment. The closed-loop response in reaction to the set point or disturbance step change 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

$$\min_{\alpha \in \mathcal{A}} = Q(\boldsymbol{\alpha}), \tag{3.4}$$

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

3.3 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 (Wilson, Hutter and Deisenroth, 2018). Unlike random search and grid search optimisation techniques where past performance is not considered to locate the global optimum (Bergstra and Bengio, 2012), Bayesian optimisation makes decisions based on the performance of previously sampled parameters. Such thoughtful choices of parameter selection characterises the sample-efficient nature of Bayesian optimisation (Bull, 2011).

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 (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, De Villiers and Cilliers, 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, Zhang and Gielen, 2013), but since it is an objective to limit the number of expensive experiments, this limitation is not of concern in this work.

3.4 GAUSSIAN PROCESSES

Per definition, a Gaussian process is a collection of random samples, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams, 2006). The random samples represent the value of the objective function $Q(\boldsymbol{\alpha})$ at inputs $\boldsymbol{\alpha}$. Gaussian processes are described by their mean and covariance function and can be written as

$$Q(\boldsymbol{\alpha}) \sim \mathcal{GP}(m(\boldsymbol{\alpha}), k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')), \tag{3.5}$$

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. (2012) propose the automatic relevance determination (ARD) Matérn parameter 5/2 kernel as the covariance function. This function is used in this work.

To demonstrate Gaussian processes visually, consider the single-dimension regression problem in Fig. 3.2 where the input α map to the output $Q(\alpha)$. Fig. 3.2 shows five random samples drawn from the prior distribution over functions specified by a Gaussian process. The shaded region in Fig. 3.2 represents the uncertainty of the sample functions by computing the variance at each input.

Gaussian processes learn the input-output relationships from a training dataset. For the problem statement defined in (3.3) and (3.4), 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

$$\widehat{Q} = Q(\boldsymbol{\alpha}) + \varepsilon \tag{3.6}$$

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

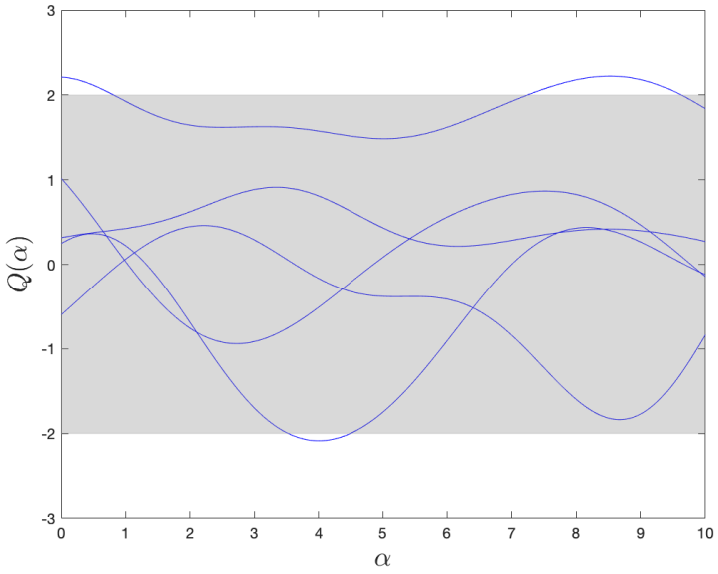

Figure 3.2. Random samples drawn from the prior distribution over the objective function.

variance σ_n^2

$$\varepsilon \sim \mathcal{N}(0, \sigma_n^2). \tag{3.7}$$

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

$$\begin{bmatrix} \widehat{\boldsymbol{Q}} \\ \boldsymbol{Q}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(A,A) + \sigma_n^2 I & K(A,A_*) \\ K(A_*,A) & K(A_*,A_*) \end{bmatrix}\right), \tag{3.8}$$

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.

$$\boldsymbol{Q}_*|A, \widehat{\boldsymbol{Q}}, A_* \sim \mathcal{N}(\bar{\boldsymbol{Q}}_*, \text{cov}(\boldsymbol{Q}_*)) \tag{3.9}$$

where,

$$\bar{\boldsymbol{Q}}_* = \boldsymbol{k}_*^\top [K + \sigma_n^2 I]^{-1} \widehat{\boldsymbol{Q}} \tag{3.10a}$$

$$\text{cov}(\boldsymbol{Q}_*)) = \boldsymbol{k}_{**} - \boldsymbol{k}_*^\top [K + \sigma_n^2 I]^{-1} \boldsymbol{k}_*. \tag{3.10b}$$

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

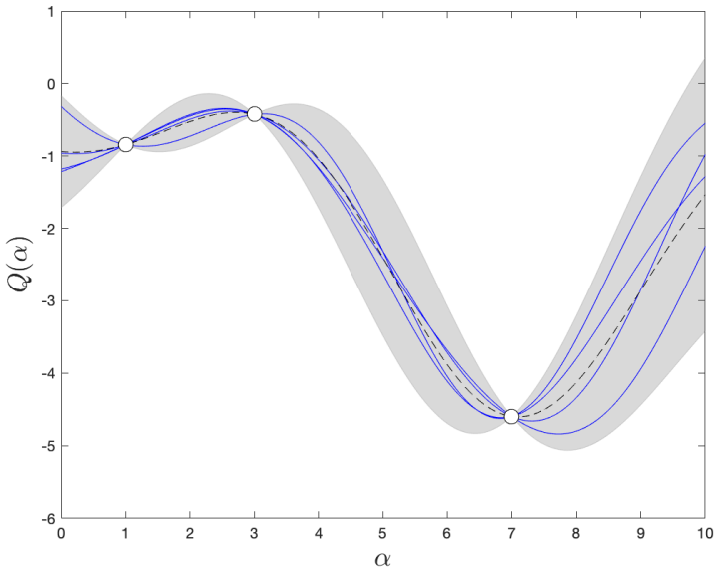

Figure 3.3. Random samples drawn from the posterior distribution conditioned at three points.

The posterior distribution of a Gaussian process combines the prior understanding of the underlying function (i.e., the mean and covariance function) with the observed data to provide an updated model and uncertainty estimates. It provides a full probabilistic representation of the underlying function, capturing both the predictions and their associated uncertainties. Fig. 3.3 shows the posterior distribution of functions, of a single dimensional regression problem, conditioned on the prior and a dataset of three data points. Only functions that pass through the points of the dataset are considered. The dashed line shows the mean prediction of the functions. The shaded region shows how the uncertainty decreases as the function approach the observations. The solid lines show four sample functions drawn from the posterior distribution.

3.5 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 (Shahriari, Swersky, Wang, Adams and De Freitas, 2015). The global minimum refers to the best possible solution of the objective function. It is the lowest value that the objective function can achieve among all possible solutions. The global minimum may be excluded from the constrained search space if the constraints are selected too conservatively to eliminate, as an example, unwanted controller behaviour.

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 (Shahriari et al., 2015).

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)

$$\boldsymbol{\alpha}_* = \underset{\boldsymbol{\alpha}\in\mathcal{A}}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{\alpha}|\mathcal{D}) \tag{3.11}$$

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 (Snoek et al., 2012). In this work EI (Mockus, 1975) is selected, as it has been shown to escape local optimums (Emmerich, Giannakoglou and Naujoks, 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

$$\mathrm{EI}(\boldsymbol{\alpha}) = \mathbb{E}\max[0, \widehat{Q}(\boldsymbol{\alpha}_{min}) - \widehat{Q}(\boldsymbol{\alpha})] \tag{3.12}$$

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 (Jones, Schonlau and Welch, 1998) as

$$\mathrm{EI}(\boldsymbol{\alpha}) = \begin{cases} (\widehat{Q}(\boldsymbol{\alpha}_{min}) - \bar{\boldsymbol{Q}}_*(\boldsymbol{\alpha}))\psi(Z) + \sigma(\boldsymbol{\alpha})\phi(Z), & \text{if } \sigma(\boldsymbol{\alpha}) > 1 \\ 0, & \text{if } \sigma(\boldsymbol{\alpha}) = 0 \end{cases} \tag{3.13}$$

where

$$Z = \frac{(\widehat{Q}(\boldsymbol{\alpha}_{min}) - \bar{\boldsymbol{Q}}_*(\boldsymbol{\alpha}))}{\sigma(\boldsymbol{\alpha})}. \tag{3.14}$$

$\sigma(\boldsymbol{\alpha})$ 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 (3.13) 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 maximizing acquisition function $\mathcal{L}$
$\boldsymbol{\alpha}_* = \underset{\boldsymbol{\alpha}\in\mathcal{A}}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{\alpha}|\mathcal{D})$
3: sample process at $\boldsymbol{\alpha}_*$ to observe $\widehat{Q}_n$
4: augment data set $\mathcal{D} = \{\mathcal{D}_{n-1}, (\boldsymbol{\alpha}_*, \widehat{Q}_n)\}$
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 results are observed and returned to augment the training dataset $\mathcal{D}$.
- The posterior distribution (surrogate model) of the objective function is updated using the augmented dataset.

This process repeats itself until a predetermined number of cycles has been reached. Refer to Algorithm 1 for the pseudocode of the process (Shahriari et al., 2015). 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.

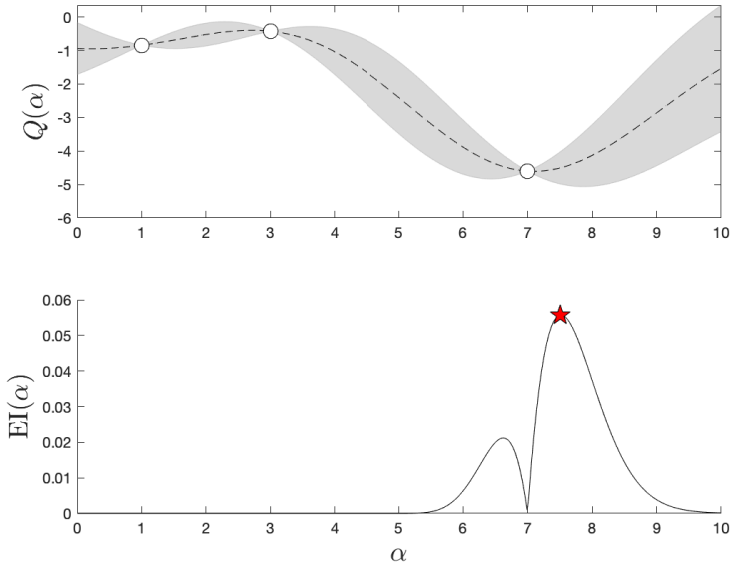

Figure 3.4. Posterior distribution and EI acquisition function indicating the next input location to be sampled with an asterix.

Fig. 3.4 shows a posterior distribution of a single dimensional regression problem conditioned on a dataset of three data points. The attributes of the posterior distribution are combined in the EI acquisition function and maximised to find α_*.

CHAPTER 4 AUTO-TUNING OF SURGE TANK CONTROLLERS

4.1 CHAPTER OVERVIEW

This chapter presents the approach followed to auto-tune MIMO controllers of a BTT surge tank. The feasibility of the approach is first tested on a linear plant model with a decentralised PI controller. The objective of the feasibility study is to determine if Bayesian optimisation can automatically tune and improve the performance of a MIMO controller. Objective functions are designed to optimise set point tracking and disturbance rejection of the controller. Once feasibility is established the approach is applied to a more complex non-linear plant model with a multi-variable controller. The response of the controllers are compared against benchmark controllers from Rokebrand et al. (2021).

4.2 BULK TAILINGS TREATMENT SURGE TANK

Fig. 4.1 taken from Rokebrand et al. (2021) and Burchell, le Roux and Craig (2023) 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 kept 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 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.

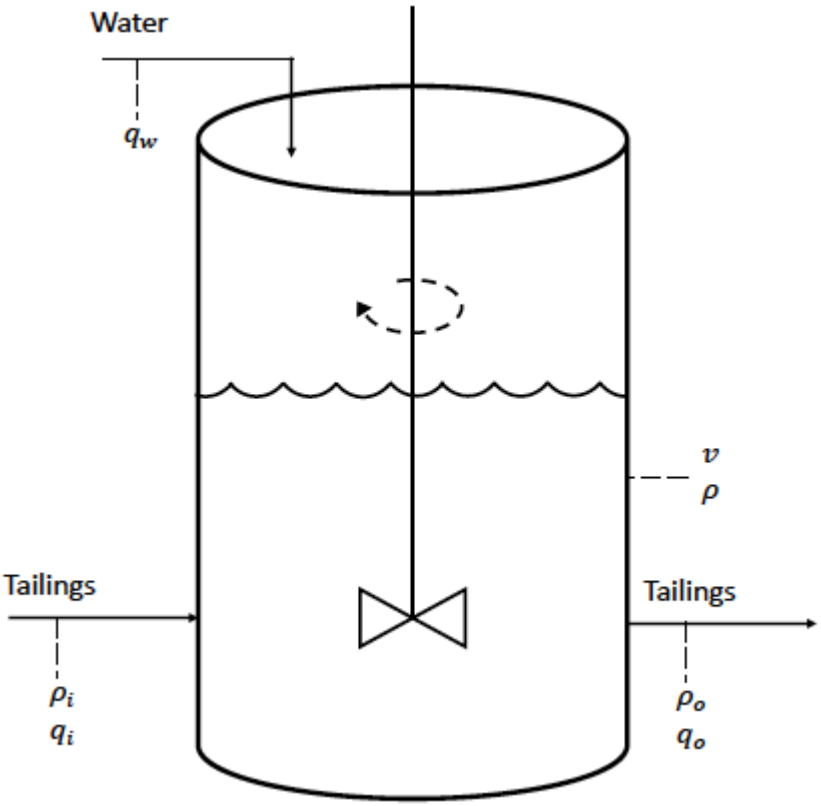

Figure 4.1. Bulk tailings treatment surge tank.

4.3 BULK TAILINGS TREATMENT SURGE TANK PLANT MODEL

Rokebrand et al. (2021) derives the BTT surge tank dynamic model from mass balance equations and presents the non-linear state space model as

$$\begin{bmatrix} \dot{v} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} q_i + q_w - q_o \\ \frac{1}{v}(\rho_i q_i + q_w - \rho(q_i + q_w)) \end{bmatrix} \tag{4.1}$$

which is of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}). \tag{4.2}$$

Equation 4.1 is linearised around the equilibrium point where the flow input equals the flow output and the surge tank volume remains constant. The linearised state space equation is converted to a transfer function matrix model of the form

$$\boldsymbol{y} = \boldsymbol{G}_p(s)\boldsymbol{u} + \boldsymbol{G}_d(s)d, \tag{4.3}$$

presented by

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s} \\ \frac{0.01}{s+75} & \frac{-0.04}{s+75} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{60}{s+75} \end{bmatrix} d. \tag{4.4}$$

Outputs y_1 and y_2 correspond to v and ρ and inputs u_1 and u_2 correspond to q_i and q_w. Table 4.1 lists the nominal, minimum and maximum values of the variables. The controlled variables must be maintained between the operational bounds given and the manipulated variables must remain within the saturation limits.

Table 4.1. Bulk tailings treatment variable descriptions.

Variable	Description	Unit	Min	Max	Nominal Value
Controlled variables					
v	Surge tank volume	m^3	3	20	10
ρ	Surge tank density	t/m^3	1	1.5	1.4
Manipulated variables					
q_i	Tailings supply flow rate	m^3/h	300	1200	600
q_w	Water flow rate	m^3/h	0	750	150
Disturbance variables					
ρ_i	Chrome tailings density	t/m^3	1	2	1.5

4.4 AUTO-TUNING OF THE BTT SURGE TANK DECENTRALISED PI CONTROLLER

4.4.1 Decentralised PI controller

The surge tank is controlled in closed-loop by controller $\boldsymbol{K}_{dc}$ tuned by Rokebrand et al. (2021) using SIMC tuning rules (Skogestad, 2003)

$$\boldsymbol{K}_{dc} = \begin{bmatrix} \frac{84(s+50)}{s} & 0 \\ 0 & \frac{-1505.7(s+75)}{s} \end{bmatrix}. \tag{4.5}$$

Controller $\boldsymbol{K}_{dc}$ is structured as a decentralised controller with PI controllers on the diagonal

$$\boldsymbol{K}_{dc} = \begin{bmatrix} k_{11} & 0 \\ 0 & k_{22} \end{bmatrix}. \tag{4.6}$$

The PI controllers in the Laplace domain are of the form (Seborg et al., 2011)

$$k_{jj} = k_{Pjj}(1 + \frac{1}{\tau_{Ijj}s}), j = 1,2 \tag{4.7}$$

where k_P is the proportional gain and τ_I is the integral time. The objective is to auto-tune controller $\boldsymbol{K}_{dc}$ by means of Bayesian optimisation to find the optimal tuning parameters k_{P11}, τ_{I11}, k_{P22} and τ_{I22}. Pre-requisites of Bayesian optimisation to meet the stated objective include defining the constraints of the search domain and design of an objective function to capture the optimal performance requirements of the controller. These pre-requisites are discussed in the following sections.

4.4.2 Constraints

Bayesian optimisation is a constrained regression process, and the constraints must be carefully considered, in conjunction with the objective function, for the Bayesian optimisation process to deliver useful results. The constraints define the search domain $\mathcal{A} \subseteq \mathbb{R}^d$. For the purposes of this simulation the constraints are the ranges of the tuning parameters within which the Bayesian optimisation algorithm must search for the optimal tuning parameters to minimize the objective function.

A benefit of the Bayesian optimisation process is that a controller can be optimally tuned without a process model, but without a process model the selection of the constraints is not trivial. In cases where an existing controller needs to be re-tuned due to poor performance the existing tuning parameters can be used as a point of departure and the constraints can be selected around that point. However for a controller to be commissioned in a newly constructed plant, such existing parameters are not available and a pragmatic approach is required to determine a reasonable search domain.

Plant permitting an open loop step test can be conducted to determine the magnitude and direction of the process gain as well as the time constant. 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 and not the optimal set of parameters. When applying methods such as SIMC, the selection of constraints should consider that the calculated controller proportional gain will be closer to the upper boundary than the lower boundary. The inverse will be true of the controller integral time. To curb process

instabilities during on-line optimisation, the search space should therefore not be symmetrical around the nominal tuning parameters, but rather off-centre, favouring areas where parameters are not close to an instability boundary.

To expand the search space around the nominal 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. Selection of the constraints is therefore a trade-off between selecting a small search space that could possibly exclude the global minimum and a large search space that could include unstable iterations or iterations that exceed the operational bounds. One can therefore not recklessly define a large search domain and hope that Bayesian optimisation finds the global minimum of the objective function.

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

$$k_{P11} \in [16.8, 168] \tag{4.8}$$

$$\tau_{I11} \in [0.01, 0.1] \tag{4.9}$$

$$k_{P22} \in [-3010, -301] \tag{4.10}$$

$$\tau_{I22} \in [0.0067, 0.067]. \tag{4.11}$$

A safe Bayesian optimisation algorithm SafeOPT has been developed by Sui, Gotovos, Burdick and Krause (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 Bayesian optimisation 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.

4.4.3 Objective function

The purpose of objective function $Q_{BTTtrack_1}$ is to auto-tune controller $\boldsymbol{K}_{dc}$ for improved set point tracking while minimising the interaction with the non-stepped controlled variable. Objective function $Q_{BTTtrack_1}$ in the form of (3.3) is

$$Q_{BTTtrack_1} = \omega_1(\beta_{11}q_1 + \beta_{12}q_2) + \omega_2(\beta_{23}q_3 + \beta_{24}q_4). \tag{4.12}$$

For the stated purpose, settling time and peak response are selected as the performance indices. Incorporating both settling time and peak response in the design of the objective function is expected to reduce the overshoot that occurs after a step change and decrease the time required for the response to settle.

$Q_{BTTtrack_1}$ consist of two terms. The first term represents the tank volume and the second the tank density. $Q_{BTTtrack_1}$ evaluates the response of two controlled variables, therefore a single Bayesian optimisation iteration of $Q_{BTTtrack_1}$ will consist of two set point step changes.

Performance index q_1 is the settling time of the surge tank volume in response to a volume set point change. Performance index q_2 is the peak response of the tank volume in response to a density set point change. Performance index q_3 is the settling time of the surge tank density in response to a density set point change. Performance index q_4 is the peak response of the tank density in response to a volume set point change. 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 toward the training dataset $\mathcal{D}$, i.e. a wasted iteration.

The performance indices of $Q_{BTTtrack_1}$ are normalised so that their magnitudes are comparable in contribution the objective function value. Normalisation is achieved by scaling each of the performance indices. The performance results of controller $\boldsymbol{K}_{dc}$ are used as the scaling factors of $Q_{BTTtrack_1}$. To obtain these results, setpoint step changes of both the volume and density setpoints are conducted, and the resulting peak magnitudes and settling times are measured. To normalise or scale the magnitudes of the performance criteria, β is selected to be the inverse of the measured performance results. Scaling factor $\beta_{11} = 1/0.094458$ is the inverse of the volume settling time of controller $\boldsymbol{K}_{dc}$ in response to

a volume set point change. $\beta_{12} = 1/0.587158$ is the inverse of the tank volume peak response to a density set point change. $\beta_{23} = 1/0.039192$ is the inverse of the density settling time in response to a density set point change. $\beta_{24} = 1/0.006402$ is the inverse of the tank density peak response to a volume set point change.

The performance weights ω_1 and ω_2 are selected to penalise a particular output to promote a favourable response. The process can tolerate slow settling times of the tank volume since there is no economic benefit for quick settling times of the volume, however a stable density supply to the concentrator improves separation efficiencies. For that reason $\omega_1 = 1$ and $\omega_2 = 2$.

Objective function $Q_{BTTreject_1}$ in the form of (3.3) is

$$Q_{BTTreject_1} = \omega_1 q_1 + \omega_2 q_2. \tag{4.13}$$

Objective function $Q_{BTTreject_1}$ is designed to improve the disturbance rejection properties of $\boldsymbol{K}_{dc}$. Transient time is selected as the performance index. Incorporating transient time in the design of the objective function is expected to decrease the time required for the transient dynamics to decay. Simulation results revealed that incorporating the peak response did not lead to improved performance and was therefore omitted for the sake of simplicity. Similar to settling time, the evaluation period of transient time must also be long enough to allow transient dynamics to decay.

$Q_{BTTreject_1}$ consist of two terms representing the transient times of the tank volume and the tank density respectively. Both terms are affected by the same disturbance input, therefore only a single step change of the disturbance input is required per Bayesian optimisation iteration. For this reason it is expected that the total optimisation time for $Q_{BTTreject_1}$ will be less than the time required to optimise $Q_{BTTtrack_1}$. As for objective function $Q_{BTTtrack_1}$, $\omega_1 = 1$ and $\omega_2 = 2$.

The transient times of the two terms are of similar magnitude, therefore $Q_{BTTreject_1}$ does not require any scaling.

The period of the iterations is determined by the choice of objective function. For the objective functions (4.12) and (4.13), the tuning parameters cannot converge at a rate faster than the time it takes for the process dynamics to die out and remain within the 2% tolerance region. The evaluation period must be sufficiently long for the performance index to be measurable so that the objective function

can return a valid scalar quantity to update the posterior distribution of the surrogate model. The evaluation period is selected to be 36 minutes per step test for both objective functions. The simulation demonstrated that the optimisation did not converge for evaluation periods less than 30 minutes due to a significant number of iterations failing to settle within the allowed time.

4.4.4 Auto-tuning for improved set point tracking using objective function $Q_{BTTtrack_1}$

4.4.4.1 Procedure

Figure (4.2) shows the Bayesian optimisation framework used for auto-tuning (Neumann-Brosig et al., 2020). For the feasibility tests, closed-loop tests are conducted on the linear model of (4.4) by stepping the set points of the controller. To evaluate set point step changes, two step tests are required, one for the volume set point and the other for the density set point, in a single Bayesian optimisation iteration. The magnitudes of the set point step changes are selected to be 2 m^3 for volume and -0.05 t/m^3 for density. A negative density step is required to prevent the maximum limit of density from being exceeded. The control errors $\mathbf{e}(t)$, which are the difference between the reference set points $\mathbf{r}(t)$ and controlled variables $\mathbf{y}(t)$, are provided as input to the feedback controller $\boldsymbol{K}_{BTTtrack_1}$.

The objective of the controller is to adjust the manipulated variables $\mathbf{u}(t)$ to keep the control error small despite changes to the set points $\mathbf{r}(t)$. The performance of the next set of controller tuning parameters $\boldsymbol{\alpha}_*$ selected by the acquisition function is evaluated in closed-loop operation and observed through the objective function $Q_{BTTtrack_{1n}}$, where n is the instance of the iteration. The tuning parameters and the observed objective function value are appended to the training dataset $\mathcal{D}$. The posterior surrogate model is updated with training set data and the acquisition function selects the next set of tuning parameters $\boldsymbol{\alpha}_{*+1}$ to be evaluated in search of the objective function minimum.

4.4.4.2 Simulation

Fig. 4.3 shows the results of the tank volume set point step change using objective function $Q_{BTTtrack_1}$ and illustrates how Bayesian optimisation explores the search space to minimise the objective function. Each trend in Fig. 4.3 represents a closed-loop step response of a candidate controller with tuning parameters selected by the acquisition function. The tank volume is shown to increase to meet the set point step change demand. The density response confirms that interaction exists between the manipulated variables controlling the volume and the tank density. The step response of the best performing controller $\boldsymbol{K}_{BTTtrack_1}$ that has minimised the objective function is highlighted. The controlled variables remain within operational bounds listed in Table 4.1 during the optimisation process.

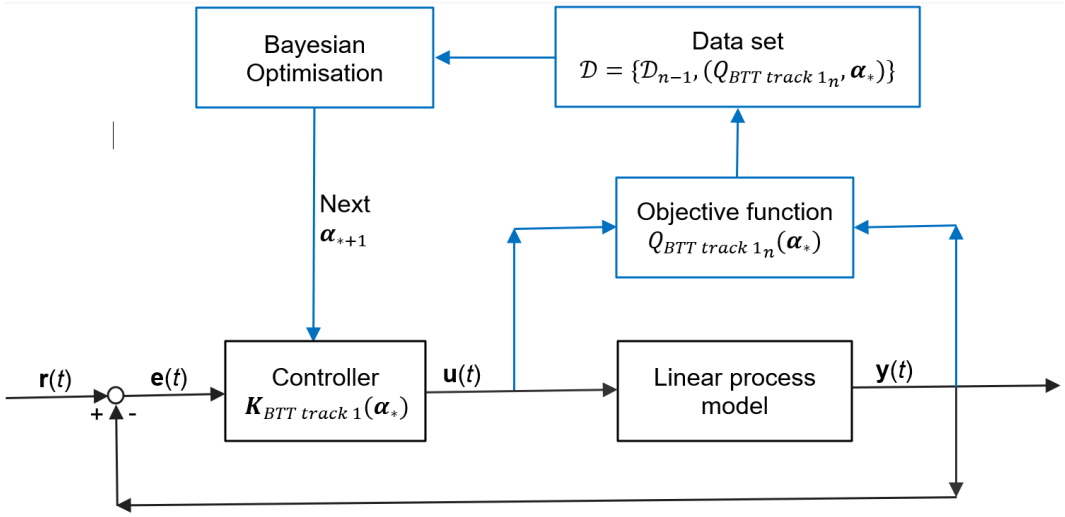

Figure 4.2. Bayesian optimisation framework to to auto-tune $\boldsymbol{K}_{BTTtrack_1}$. The blue elements represent the required adaptation of the unitary feedback controller block diagram to implement Bayesian optimisation.

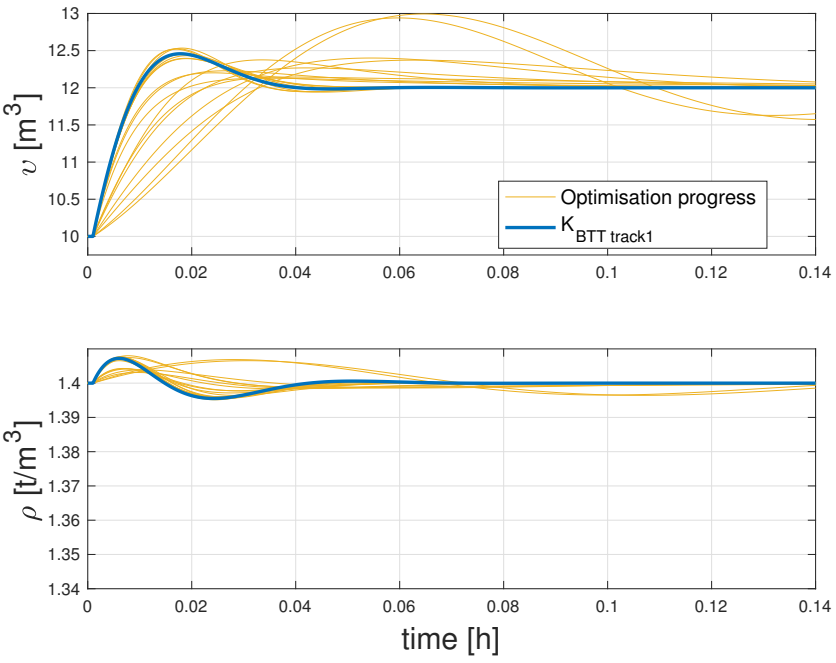

Figure 4.3. Response of the controlled variables to a volume set point step change during Bayesian optimisation using objective function $Q_{BTTtrack_1}$.

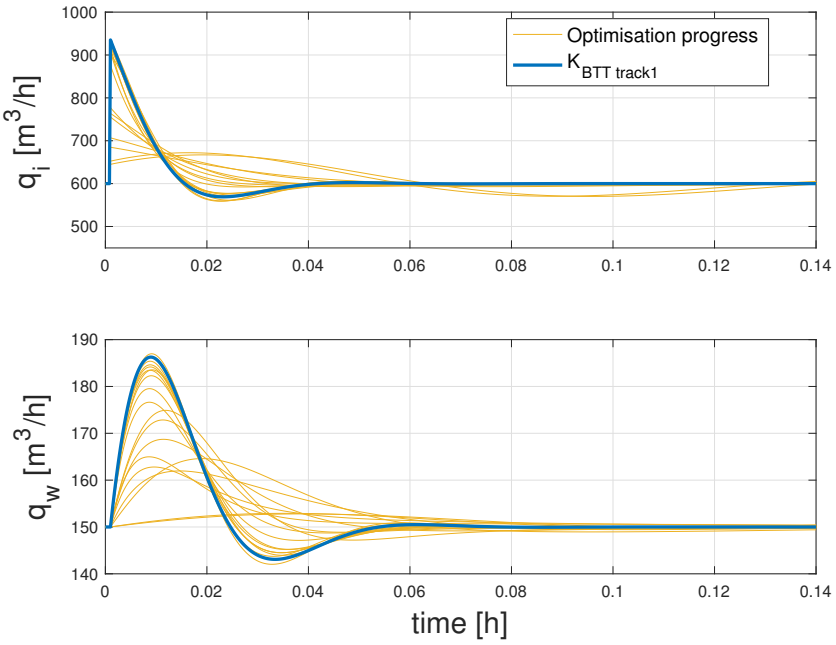

Figure 4.4. Response of the manipulated variables to a volume set point step change during Bayesian optimisation using objective function $Q_{BTTtrack_1}$.

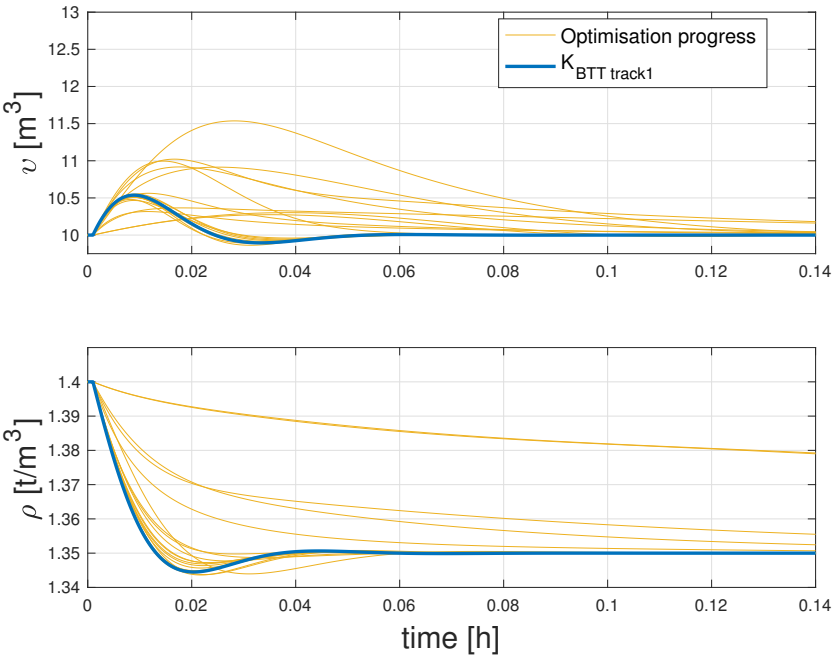

Figure 4.5. Response of the controlled variables to a density set point step change during Bayesian optimisation using objective function $Q_{BTTtrack_1}$.

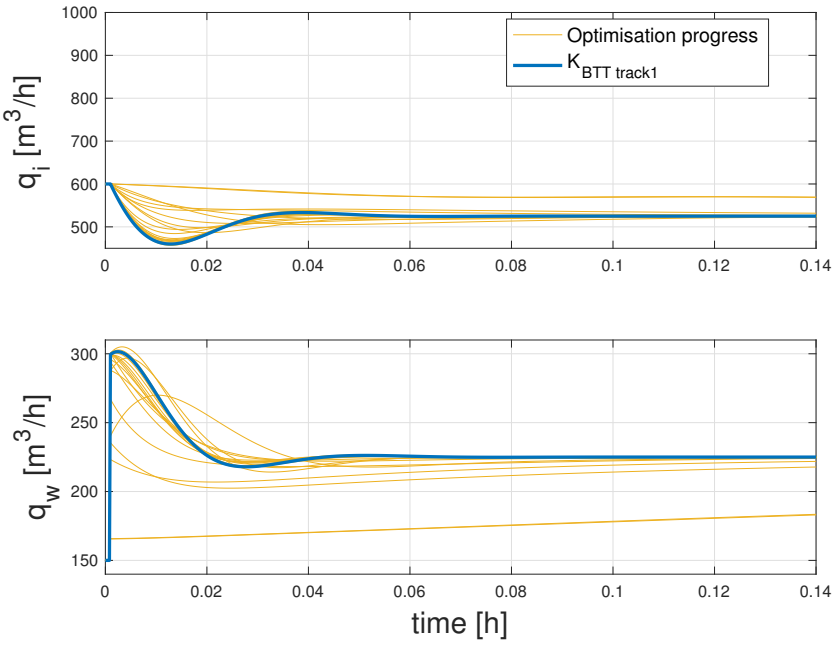

Figure 4.6. Response of the manipulated variables to a density set point step change during Bayesian optimisation using objective function $Q_{BTTtrack_1}$.

Fig. 4.4 shows how the manipulated variables respond to the volume set point step change. The tailings supply flow immediately increases in response to the volume set point step change to increase the tank volume. The increase of the tailings supply interacts with the tank density and causes the density to increase. The water flow rate increases to counteract the increasing tank density. Both manipulated variables remain within the variable constraints listed in Table 4.1 during the optimisation process.

Fig. 4.5 shows the results of the tank density ρ set point step change using objective function $Q_{BTTtrack_1}$. The tank density is shown to decrease to meet the set point step change demand. The volume response confirms that interaction exists between the manipulated variables controlling the density and the tank volume.

Fig. 4.6 shows how the manipulated variables respond to the density set point step change. The water supply flow immediately increases in response to the density set point step change to decrease the tank density. The increase of the water supply interacts with the tank volume and causes the volume to increase. The tailing supply flow decreases to stabilise the tank volume.

Table 4.2 shows the results of the Bayesian optimisation simulation using $Q_{BTTtrack_1}$, iterations 11 through to 20. Column $Q_{BTTtrack_1}$ represents the objective function value for each set of tuning parameters applied. The no results entries are due to the controlled variable responses not settling within evaluation period. The best result is found by iteration 20 with optimal tuning parameters as shown in the corresponding row.

Table 4.2. Results of the Bayesian optimisation simulation to improve set point tracking using $Q_{BTTtrack_1}$, iterations 11 through 20.

Iter	$Q_{BTTtrack_1}$	k_{P11}	τ_{I11}	k_{P22}	τ_{I22}
11	5.2506	167.47	0.01661	-2997	0.007359
12	no result	161.93	0.01426	-2880	0.012868
13	5.3324	162.16	0.01043	-2990	0.007564
14	5.2475	159.44	0.01333	-2915	0.009586
15	5.3335	151.06	0.01055	-3001	0.009818
16	no result	26.196	0.01313	-315	0.064597
17	5.2934	167.76	0.01679	-2977	0.008090
18	no result	22.488	0.01290	-310	0.064967
19	5.3373	167.58	0.01016	-3001	0.009227
20	**5.1937**	**167.58**	**0.01305**	**-2990**	**0.007202**

During simulation, the step test response was evaluated over a period of 36 minutes in order for the process dynamics to die out. To conduct 20 Bayesian optimisation iterations as suggested in Table 4.2, with each iteration requiring two step tests would require a total of 40 step tests in practice. The optimisation process would take no less than 24 hours to complete.

Fig. 4.7 compares the volume step change response of controllers $\boldsymbol{K}_{BTTtrack_1}$ and $\boldsymbol{K}_{dc}$. The objective function $Q_{BTTtrack_1}$ was selected to improve settling time while minimising the interaction with the non-stepped controlled variable. The volume settling time is seen to decrease from 0.094 to 0.035 hours which is an improvement of 63%. Even though the transient time of the non-stepped output is not evaluated by the objective function, the transient time of the tank density does decrease from 0.121 hours to 0.064 hours. The interaction peak increases from 0.0064 to 0.0072 t/m^3, a deterioration of -13% (Recall that the peak is defined as the maximum value of $|y(t) - y_{init}|$ as defined in Table 3.1). Simulations show that it is possible to improve on the interaction peak by further penalising the density

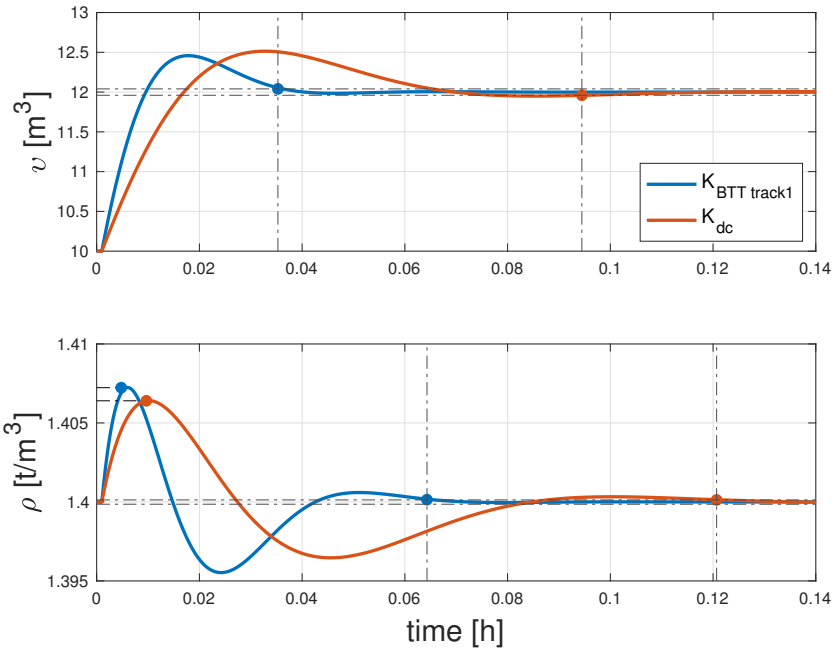

Figure 4.7. Comparison of the volume step change response of controllers $\boldsymbol{K}_{BTTtrack_1}$ and $\boldsymbol{K}_{dc}$. The markers indicate the improvement of the set point tracking settling time, interaction peak and interaction transient time.

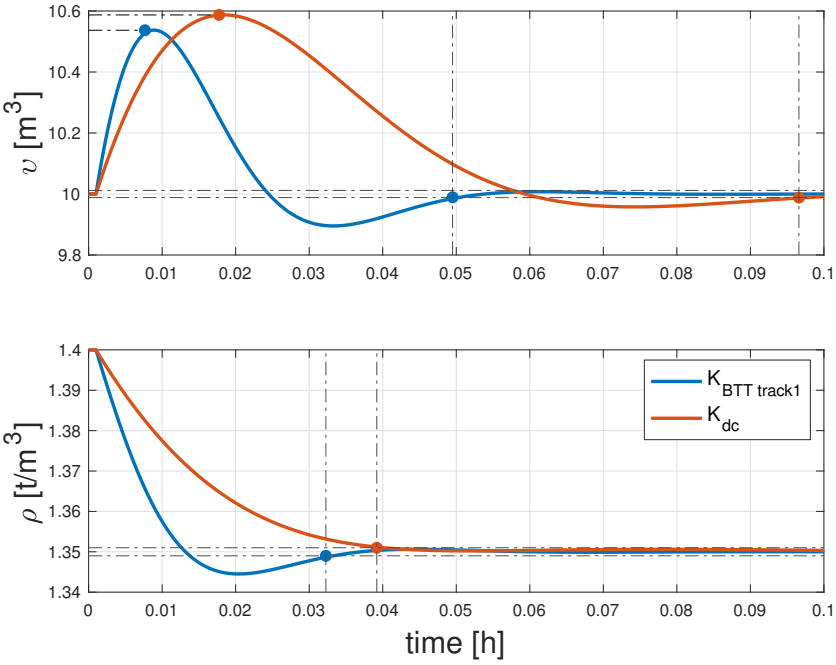

Figure 4.8. Comparison of the density step change response of controllers $\boldsymbol{K}_{BTTtrack_1}$ and $\boldsymbol{K}_{dc}$. The markers indicate the improvement of the set point tracking settling time, interaction peak and interaction transient time.

term $Q_{BTTtrack_1}$ i.e. changing $\omega_2 = 4$. The penalty improves the interaction peak at the expense of volume set point tracking performance. It was decided to favour set point tracking performance over an improvement of the interaction peak, since it is the primary purpose of the objective function. A summary of the performance criteria results is provided in Table 4.3.

Fig. 4.8 compares the density step change response of controllers $\boldsymbol{K}_{BTTtrack_1}$ and $\boldsymbol{K}_{dc}$. The density settling time is seen to decrease from 0.039 to 0.032 hours which is an improvement of 18%. The transient time of the tank volume decreases from 0.097 hours to 0.049 hours, and the interaction peak reduces slightly from 0.58 to 0.54 m^3, which is an improvement of 9%.

Table 4.3. Comparison of performance criteria of controllers $\boldsymbol{K}_{dc}$ and $\boldsymbol{K}_{BTTtrack_1}$.

Performance criteria	Unit	$\boldsymbol{K}_{dc}$	$\boldsymbol{K}_{BTTtrack_1}$	Improvement (%)
Volume settling time	hours	0.09445	0.03528	62.65
Density settling time	hours	0.03919	0.03227	17.66
Volume peak interaction	m^3	0.58715	0.53695	8.55
Density peak interaction	t/m^3	0.00640	0.00723	-12.97
Volume step RMSE		0.37371	0.28297	24.28
Density interaction RMSE		0.00203	0.00173	14.82
Volume interaction RMSE		0.20495	0.12652	38.27
Density step RMSE		0.01016	0.00733	27.77

The Root Mean Squared Error (RMSE) calculated in Table 4.3 provides a statistical measure of performance improvement. The results from Table 4.3 show that Bayesian optimisation has improved the set point tracking performance of $\boldsymbol{K}_{BTTtrack_1}$ within the constraints provided using the objective function $Q_{BTTtrack_1}$. The approach implemented to auto-tune $\boldsymbol{K}_{BTTtrack_1}$ for improved set point tracking is feasible and suitable to be implemented on a non-linear plant model and more complex multi-variable controller.

4.4.5 Auto-tuning for improved disturbance rejection using objective function $Q_{BTTreject_1}$

4.4.5.1 Procedure

Figure (4.9) shows the framework within which controller $\boldsymbol{K}_{BTTreject_1}$ is auto-tuned for improved disturbance rejection. For the feasibility tests, closed-loop tests are conducted on the linear model of (4.4) by stepping the disturbance input. To evaluate disturbance step changes, a single step test

is required for every Bayesian optimisation iteration because there is ony one disturbance input. To compare the results of the optimised controller performance with that of Rokebrand et al. (2021) a disturbance step change of 0.1 t/m^3 is selected. The control errors $\mathbf{e}(t)$, which are the difference between the reference set points $\mathbf{r}(t)$ and controlled variables $\mathbf{y}(t)$, are provided as input to the feedback controller $\boldsymbol{K}_{BTTreject_1}$.

The objective of the controller is to adjust the manipulated variables $\mathbf{u}(t)$ to keep the control error small despite the disturbances $\mathbf{d}(t)$. The performance of the next set of controller tuning parameters $\boldsymbol{\alpha}_*$ selected by the acquisition function is evaluated in closed-loop operation and observed through the objective function $Q_{BTTreject_{1n}}$, where n is the instance of the iteration. The tuning parameters and the observed objective function value are appended to the training dataset $\mathcal{D}$. The posterior surrogate model is updated with training set data and the acquisition function selects the next set of tuning parameters $\boldsymbol{\alpha}_{*+1}$ to be evaluated in search of the objective function minimum.

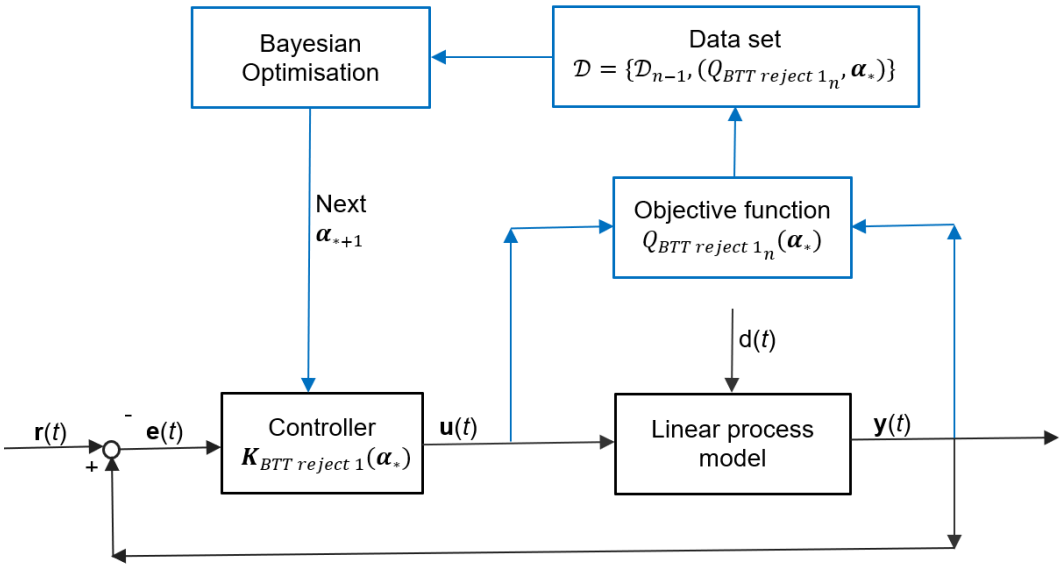

Figure 4.9. Bayesian optimisation framework to to auto-tune $\boldsymbol{K}_{BTTreject_1}$. The blue elements represent the required adaptation of the unitary feedback controller block diagram to implement Bayesian optimisation.

4.4.5.2 Simulation

Fig. 4.10 shows the volume and density response during the Bayesian optimisation process for a step disturbance of 0.1 t/m^3 in the Chrome tailings density. The step response of the best performing controller $\boldsymbol{K}_{BTTreject_1}$ evaluated using objective fuction $Q_{BTTreject_1}$, is highlighted.

Fig. 4.11 shows how the controller manipulates the tailings supply and water flow rates to reject the disturbance. Due to the increase of the Chrome tailings density, the surge tank density increases. In response to the increased surge tank density, the controller increases the water flow rate which suppresses the density disturbance, but also increases the surge tank level. The controller responds by decreasing the tailings supply flow rate to stabilise the tank level. Both the controlled variables and manipulated variables remain within process bounds during the optimisation process.

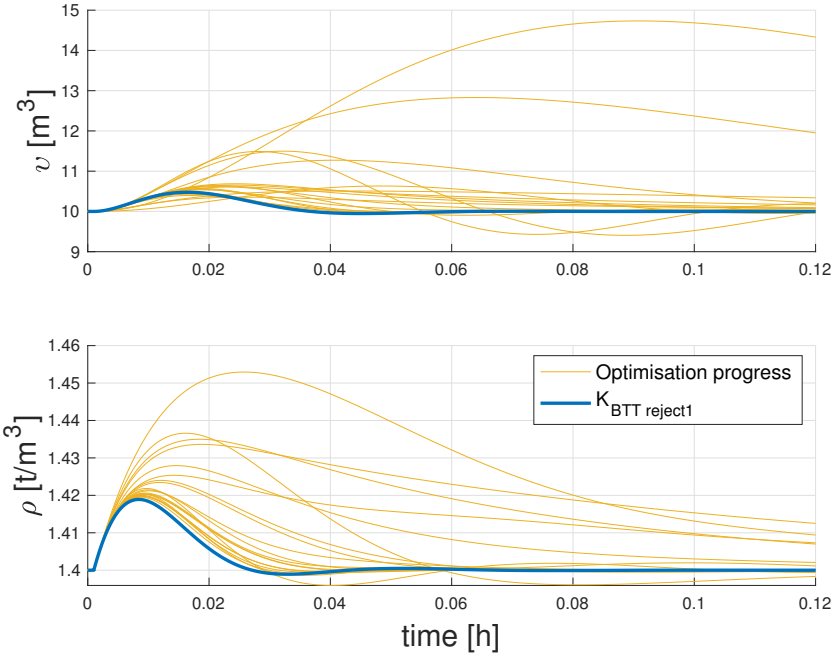

Figure 4.10. Response of the controlled variables to a disturbance step change during Bayesian optimisation using objective function $Q_{BTTreject_1}$.

Table 4.4 shows the results of the Bayesian optimisation simulation using objective function $Q_{BTTreject_1}$, iterations 11 through to 20. The objective function is sufficiently minimised by iteration no. 15 to provide an acceptable margin of improvement upon the performance of the benchmark controller $\boldsymbol{K}_{dc}$. Note that the iterations do not stop once the objective function minimum is located, but continues until the pre-set number of iterations are complete. Objective function $Q_{BTTreject_1}$ only requires a single disturbance step test to evaluate. At 36 minutes per test, the 20 step tests would require no less than 12 hours to complete in practice. As anticipated the total optimisation time for $Q_{BTTreject_1}$ is less than that for $Q_{BTTtrack_1}$.

To evaluate the performance of the controller optimised for disturbance rejection, the controller is benchmarked against results from Rokebrand et al. (2021). Fig. 4.12 compares the disturbance

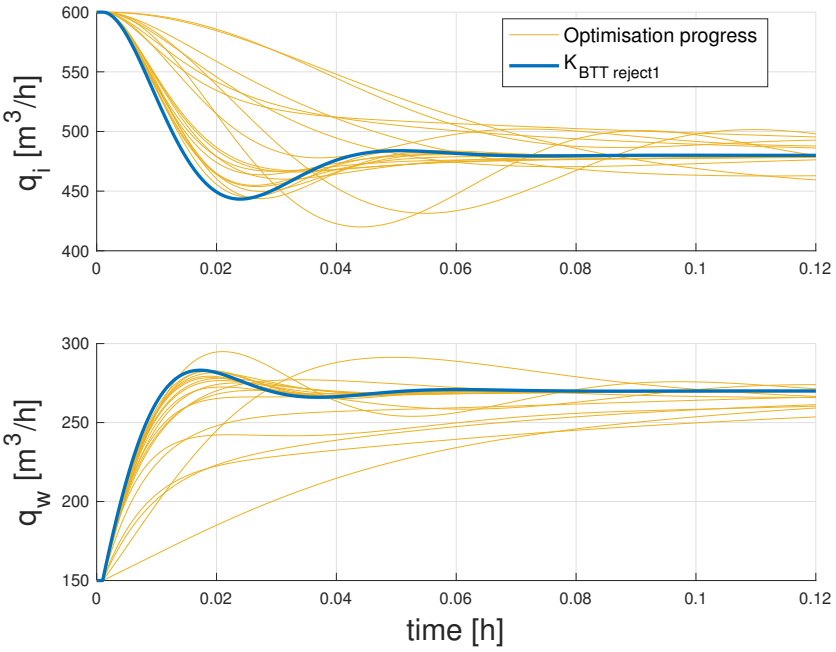

Figure 4.11. Response of the manipulated variables to a disturbance step change during Bayesian optimisation using objective function $Q_{BTTreject_1}$.

Table 4.4. Results of the Bayesian optimisation simulation using objective function $Q_{BTTreject_1}$, iterations 11 through 20.

Iter	$Q_{BTTreject_1}$	k_{P11}	τ_{I11}	k_{P22}	τ_{I22}
11	0.43986	29	0.010503	-2963	0.017718
12	0.11939	165	0.012518	-2982	0.013052
13	0.11951	144	0.010614	-2868	0.0076114
14	0.1239	167	0.021729	-2921	0.0071512
15	**0.11421**	**158**	**0.011173**	**-3003**	**0.0066239**
16	0.12097	159	0.01168	-2911	0.0097733
17	0.44786	34	0.010419	-303	0.010039
18	0.13112	133	0.013029	-2697	0.0069357
19	0.61284	17	0.099396	-851	0.006959
20	0.12868	138	0.014263	-3008	0.008375

rejection response of controllers $\boldsymbol{K}_{BTTreject_1}$ and $\boldsymbol{K}_{dc}$. Objective function $Q_{BTTreject_1}$ was designed to improve on the disturbance rejection transient time. The volume transient time is seen to reduce from 0.108 to 0.058 hours, a 46% improvement. The density transient time reduces from 0.109 to 0.056 hours, a 48% improvement. Fig. 4.12 also shows an improvement of the controlled variable peaks induced by the disturbance, even though peak reduction was not optimised for.

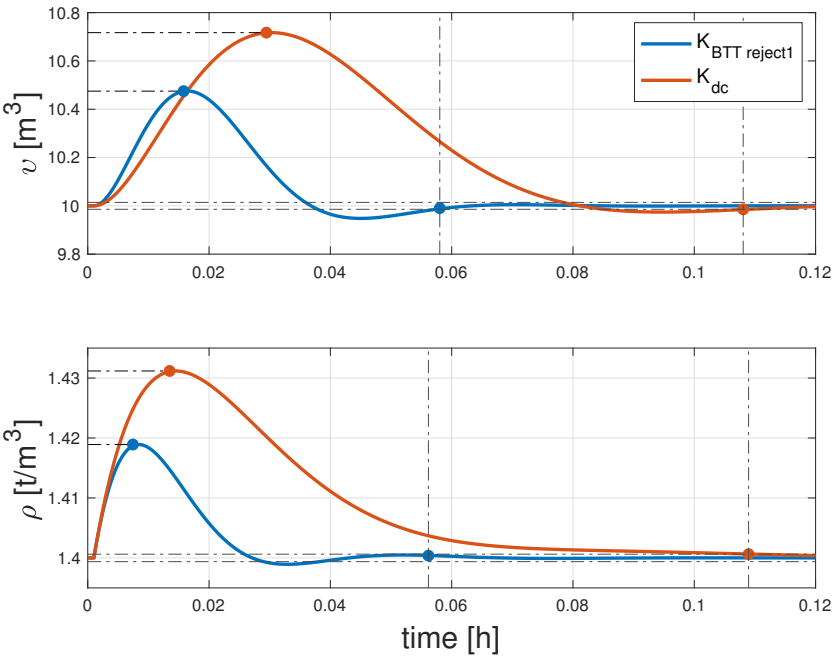

Figure 4.12. Comparison of the disturbance step change response of controllers $\boldsymbol{K}_{BTTreject_1}$ and $\boldsymbol{K}_{dc}$. The markers indicate the improvement of the disturbance peaks and transient times.

Table 4.5 summarises the key performance criteria of controller $\boldsymbol{K}_{BTTreject_1}$ and $\boldsymbol{K}_{dc}$. Included in the results are the RMSE calculations which provide a statistical comparison of performance. The results listed in Table 4.5 confirm that Bayesian optimisation with objective function $Q_{BTTreject_1}$ is successful in auto-tuning $\boldsymbol{K}_{BTTreject_1}$ to improve disturbance rejection performance. Auto-tuning controller $\boldsymbol{K}_{BTTreject_1}$ for improved disturbance rejection performance is therefore feasible and suitable to be implemented on a non-linear plant model and more complex multi-variable controller.

4.5 AUTO-TUNING OF THE BTT SURGE TANK INVERSE CONTROLLER

4.5.1 Multivariable controller

In this section it is assumed that the surge tank is controlled in closed-loop by controller $\boldsymbol{K}_{inv}$ (4.14) which is a modified inverse controller (Rokebrand, Burchell, Olivier and Craig, 2020). The modified

Table 4.5. Comparison of disturbance rejection criteria of controller $\boldsymbol{K}_{BTTreject_1}$ and $\boldsymbol{K}_{dc}$.

Performance criteria	Unit	$\boldsymbol{K}_{dc}$	$\boldsymbol{K}_{BTTreject_1}$	Improvement (%)
Volume transient time	hours	0.10806	0.05804	46.29
Density transient time	hours	0.10895	0.05616	48.45
Volume disturbance peak	m^3	0.71712	0.47495	33.77
Density disturbance peak	t/m^3	0.03117	0.01890	39.36
Volume response RMSE		0.27816	0.12867	53.74
Density response RMSE		0.01053	0.00438	58.41

inverse controller is used as reference rather than the ideal inverse controller (Rokebrand et al., 2021) because it poses a more complex optimisation problem to solve having two additional tuning parameters to optimise. The ideal inverse controller consists of two purely proportional elements acting on the volume error. Integral action is added to the two purely proportional elements by Rokebrand et al. (2020) to account for gain uncertainty.

$$\boldsymbol{K}_{inv} = 100 \begin{bmatrix} \frac{0.8(s+41.3)}{s} & \frac{20(s+75)}{s} \\ \frac{0.2(s+165)}{s} & \frac{-20(s+75)}{s} \end{bmatrix}. \tag{4.14}$$

Controller $\boldsymbol{K}_{inv}$ is structured as a multivariable controller with PI controllers k_{ij} in all the controller matrix positions

$$\boldsymbol{K}_{inv} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}. \tag{4.15}$$

Controller $\boldsymbol{K}_{inv}$ is auto-tuned by finding the optimal tuning parameters for k_{P11}, τ_{I11}, k_{P12}, τ_{I12}, k_{P21}, τ_{I21}, k_{P22} and τ_{I22}. $\boldsymbol{K}_{inv}$ has double the number of tuning parameters to optimise compared to $\boldsymbol{K}_{dc}$. It is expected that $\boldsymbol{K}_{inv}$ will require more iterations to optimise than for $\boldsymbol{K}_{dc}$ due to the increased number of tuning parameters.

4.5.2 Constraints

The constraints are determined using the same method discussed in Section 4.4.2, using the tuning parameters of Rokebrand et al. (2020) as a point of departure.

$$k_{P11} \in [16, 160] \tag{4.16}$$

$$\tau_{I11} \in [0.0121, 0.1211] \tag{4.17}$$

$$k_{P12} \in [400, 4000] \tag{4.18}$$

$$\tau_{I12} \in [0.0067, 0.0667] \tag{4.19}$$

$$k_{P21} \in [4, 40] \tag{4.20}$$

$$\tau_{I21} \in [0.0030, 0.0303] \tag{4.21}$$

$$k_{P22} \in [-4000, -400] \tag{4.22}$$

$$\tau_{I22} \in [0.0067, 0.0667]. \tag{4.23}$$

4.5.3 Objective function

The purpose of objective function $Q_{BTTtrack_2}$ is to auto-tune controller $\boldsymbol{K}_{inv}$ for improved set point tracking while minising the interaction with the non-stepped controlled variable. Objective function $Q_{BTTtrack_2}$ in the form of (3.3) is

$$Q_{BTTtrack_2} = \omega_1(\beta_{11}q_1 + \beta_{12}q_2) + \omega_2(\beta_{23}q_3 + \beta_{24}q_4). \tag{4.24}$$

The performance indices and weights of $Q_{BTTtrack_2}$ are equal to that of $Q_{BTTtrack_1}$.

Even though the purpose of objective functions $Q_{BTTtrack_1}$ and $Q_{BTTtrack_2}$ are the same, $Q_{BTTtrack_2}$ cannot be set equal to $Q_{BTTtrack_1}$ for the reason that the scaling factors will not be the same. The scaling factors of the objective functions differ due to the difference in performance characteristics, such as settling times and interaction peaks, of controllers $\boldsymbol{K}_{inv}$ and $\boldsymbol{K}_{dc}$.

The scaling factors for $Q_{BTTtrack_2}$ are determined as follows. Scaling factor $\beta_{11} = 2/0.078455$ which is the inverse of the volume settling time of controller $\boldsymbol{K}_{inv}$ in response to a volume set point change. The value of "2" in β_{11} is to prioritise settling time over minimisation of the interaction peak. $\beta_{12} = 1/0.13475$ is the inverse of the tank volume peak response to a density set point change. $\beta_{23} = 2/0.038276$ is the inverse of the density settling time in response to a density set point change. $\beta_{24} = 1/0.0067734$ is the inverse of the tank density peak response to a volume set point change.

Objective function $Q_{BTTreject_2}$ with the purpose of improving disturbance rejection in the form of (3.3) is

$$Q_{BTTreject_2} = \omega_1 q_1 + \omega_2 q_2. \tag{4.25}$$

The performance indices and weights of $Q_{BTTreject_2}$ are equal to that of $Q_{BTTreject_1}$. Unlike the set point tracking objective functions, the settling time of $\boldsymbol{K}_{inv}$ and $\boldsymbol{K}_{dc}$ are similar in magnitude. Since the magnitudes are similar and the objective function does not consider any other performance criteria other than settling time no scaling is required and $Q_{BTTreject_2} = Q_{BTTreject_1}$.

4.5.4 Auto-tuning for improved set point tracking using objective function $Q_{BTTtrack_2}$

4.5.4.1 Procedure

Closed-loop tests are conducted on the non-linear model of (4.1) by stepping the set points of the controller. The procedure as discussed in Section 4.4.4.1 applies, except that the plant is now represented by a non-linear model.

4.5.4.2 Simulation

Fig. 4.13 shows the results of the tank volume set point step change using objective function $Q_{BTTtrack_2}$ on the non-linear plant model using the multivariable inverse controller. The tank volume rises to meet the increased volume set point. The slight dip in density response confirms that interaction exists between the manipulated variables controlling the volume and the tank density controlled variable. The step response of the best performing controller $\boldsymbol{K}_{BTTtrack_2}$ that has minimised the objective function is highlighted. The controlled variables remain within operational bounds listed in Table 4.1 during the optimisation process.

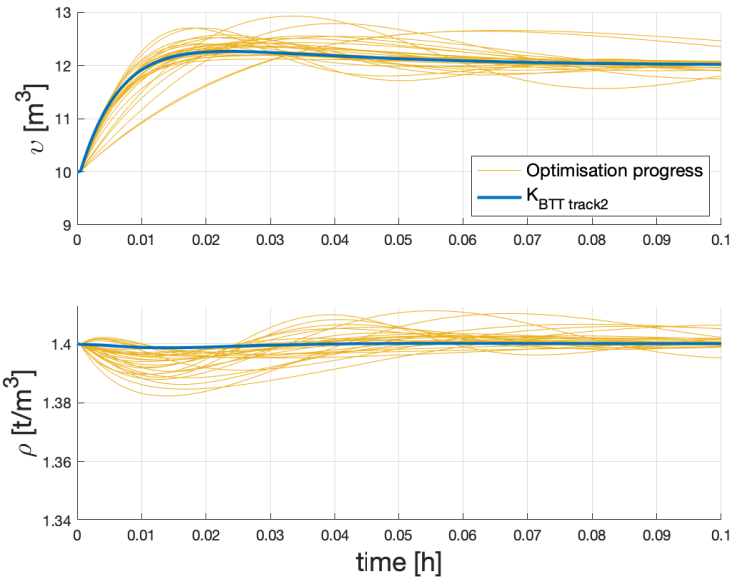

Figure 4.13. Response of the controlled variables to a volume set point step change during Bayesian optimisation using objective function $Q_{BTTtrack_2}$.

Fig. 4.14 shows how the manipulated variables respond to the volume set point step change. In contrast to the decentralised PI controller, where only the paired output responds immediately to the set point step change, both the tailings supply flow and water supply flow immediately increase in response to the volume set point step change to increase the tank volume while maintaining the density at set point. Both manipulated variables remain within the variable constraints listed in Table 4.1 during the optimisation process.

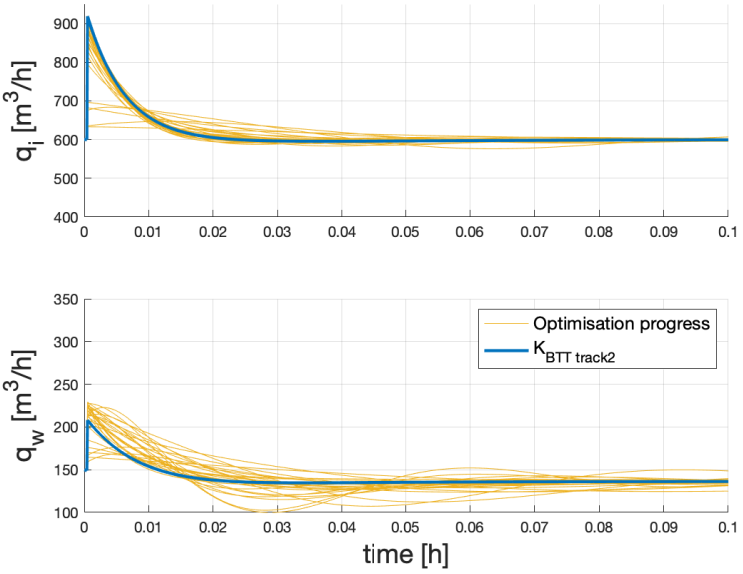

Figure 4.14. Response of the manipulated variables to a volume set point step change during Bayesian optimisation using objective function $Q_{BTTtrack_2}$.

Fig. 4.15 shows the results of the tank density set point step change using objective function $Q_{BTTtrack_2}$. The tank density is shown to decrease to meet the set point step change demand. The volume response confirms that interaction exists between the manipulated variables controlling the density and the tank volume.

Fig. 4.16 shows how the manipulated variables respond to the density set point step change. Both the tailings and water supply flow respond immediately to the change in density set point. The tailings flow decreases to reduce the tank density while the water flow increases to maintain the tank level.

Table 4.6 shows the results of the Bayesian optimisation simulation using $Q_{BTTtrack_2}$, iterations 21 through to 30. Due to the increased number of tuning parameters to optimise, the minimisation of $Q_{BTTtrack_2}$ requires substantially more iterations to converge than $Q_{BTTtrack_1}$. Column $Q_{BTTtrack_2}$

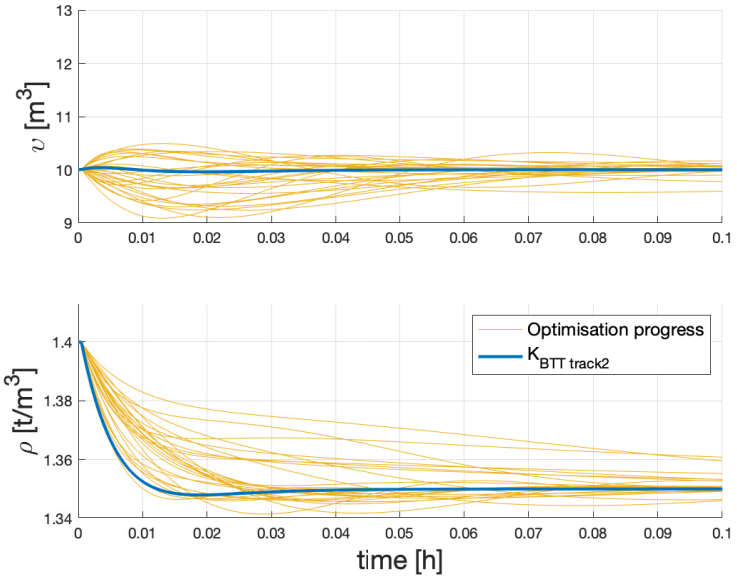

Figure 4.15. Response of the controlled variables to a density set point step change during Bayesian optimisation using objective function $Q_{BTTtrack_2}$.

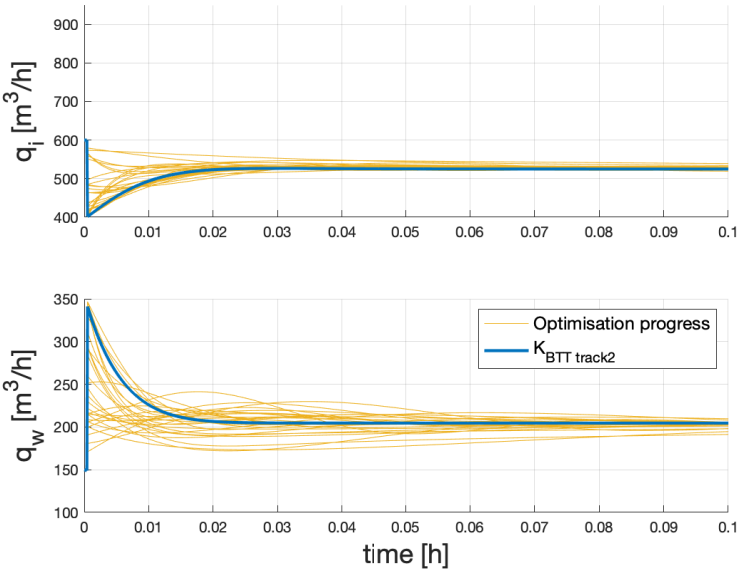

Figure 4.16. Response of the manipulated variables to a density set point step change during Bayesian optimisation using objective function $Q_{BTTtrack_2}$.

represents the objective function value for each set of tuning parameters applied. The best result is found by iteration 26 with optimal tuning parameters as shown in the corresponding row.

Table 4.6. Results of the Bayesian optimisation simulation to improve set point tracking using $Q_{BTTtrack_2}$, iterations 21 through 30.

Iter	$Q_{BTTtrack_2}$	k_{P11}	τ_{I11}	k_{P12}	τ_{I12}	k_{P21}	τ_{I21}	k_{P22}	τ_{I22}
21	13.155	120.92	0.1019	3949	0.00680	38.193	0.0049	-3821	0.0236
22	8.9497	153.3	0.1159	3443	0.00681	36.429	0.0047	-2856	0.0103
23	17.191	135.22	0.0524	2728	0.00672	37.824	0.0177	-827	0.0182
24	11.225	153.27	0.0926	3857	0.04244	10.143	0.0109	-1000	0.0072
25	8.0661	149.9	0.0244	3986	0.02693	26.24	0.0270	-3623	0.0075
26	**5.852**	**159.45**	**0.0332**	**3975**	**0.00753**	**29.744**	**0.0260**	**-3847**	**0.0105**
27	9.1409	158.01	0.0809	3307	0.00971	13.364	0.0270	-2038	0.0068
28	14.612	159.73	0.0527	3952	0.03112	31.819	0.0047	-1318	0.0106
29	7.8549	159.43	0.0424	2713	0.00897	37.677	0.0291	-3960	0.0092
30	7.7335	143.58	0.0310	2703	0.00777	25.993	0.0049	-3960	0.0068

During simulation, the step test response was evaluated over a period of 36 minutes in order for the process dynamics to die out. To conduct 30 Bayesian optimisation iterations as suggested in Table 4.6, and each iteration requiring two step tests would require a total of 60 step tests in practice. The process would take no less than 36 hours to complete.

Fig. 4.17 compares the volume step change response of the optimised controller $\boldsymbol{K}_{BTTtrack_2}$ and reference controller $\boldsymbol{K}_{inv}$. The objective function $Q_{BTTtrack_2}$ was designed to improve the settling time while minimising the interaction with the non-stepped controlled variable. The volume settling time is seen to increase marginally from 0.0785 to 0.0786 hours which is a deterioration of -0.18%. Even thought there is a deterioration in settling time, the RMSE of the response has improved by 23% due to the faster rising time and smaller overshoot. Since the response of the volume controlled variable is economically insignificant, modifications to the objective function to improve settling time was not considered. The transient time of the tank density decreases from 0.118 hours to 0.092 hours and the interaction peak improves from 0.0068 to 0.0014 t/m^3, a 83% improvement. The interaction peak improvement is of economic consequence since downstream separation efficiencies are improved when the density of the supply is stable. The results are summarised in Table 4.7.

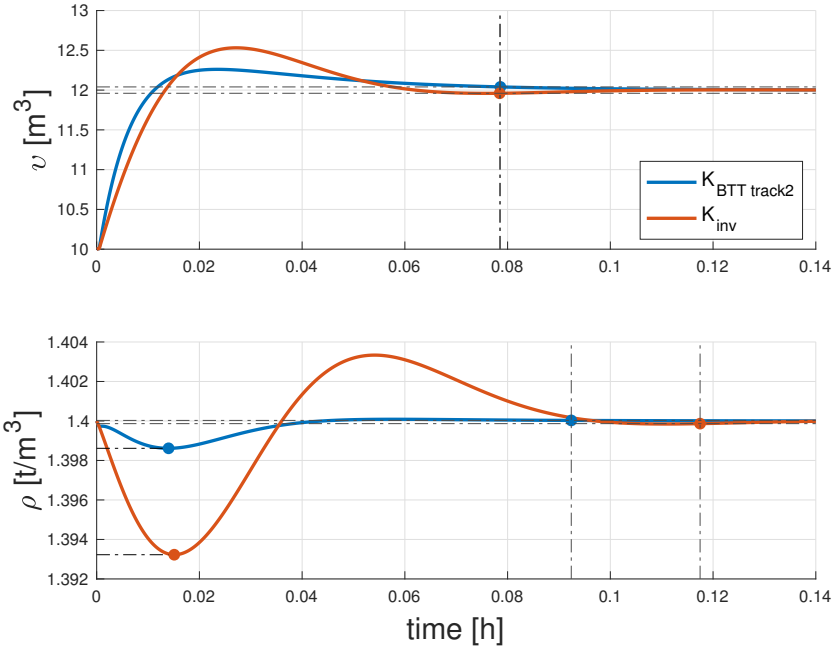

Figure 4.17. Comparison of the volume step change response of controllers $\boldsymbol{K}_{BTTtrack_2}$ and $\boldsymbol{K}_{inv}$. The markers indicate the improvement of the set point tracking settling time, interaction peak and interaction transient time.

Fig. 4.18 compares the density step change response of controllers $\boldsymbol{K}_{BTTtrack_2}$ and $\boldsymbol{K}_{inv}$. The density settling time is seen to decrease from 0.038 to 0.03 hours which is an improvement of 21%. The transient time of the tank volume decreases from 0.084 hours to 0.059 hours, and the interaction peak improves significantly from 0.13 to 0.04 m^3, an improvement of 70%.

The Root Mean Squared Error (RMSE) listed in Table 4.7 provides a statistical comparison of the performance of controllers $\boldsymbol{K}_{inv}$ and $\boldsymbol{K}_{BTTtrack_2}$. The performance criteria in Table 4.7 confirms that Bayesian optimisation with objective function $Q_{BTTtrack_2}$ is successful in auto-tuning $\boldsymbol{K}_{BTTtrack_2}$ to improve the set point tracking performance. With the exception of the volume settling time, all the performance criteria show significant improvement. The marginal deterioration of the volume settling time can be considered acceptable since it does not affect downstream processes.

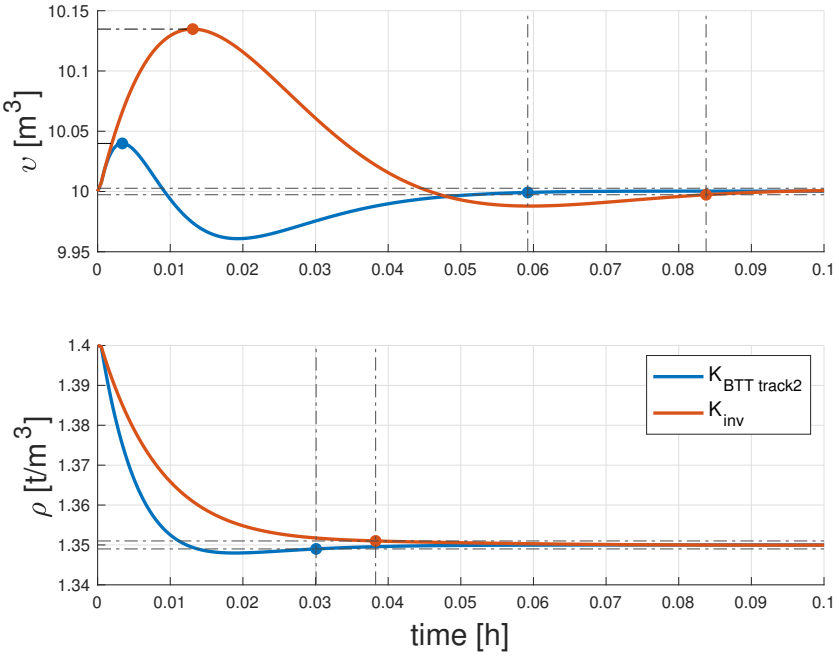

Figure 4.18. Comparison of the density step change response of controllers $\boldsymbol{K}_{BTTtrack_1}$ and $\boldsymbol{K}_{dc}$. The markers indicate the improvement of the set point tracking settling time, interaction peak and interaction transient time.

Table 4.7. Comparison of performance criteria of controllers $\boldsymbol{K}_{inv}$ and $\boldsymbol{K}_{BTTtrack_2}$.

Performance criteria	Unit	$\boldsymbol{K}_{inv}$	$\boldsymbol{K}_{BTTtrack_2}$	Improvement (%)
Volume settling time	hours	0.07845	0.07859	-0.179
Density settling time	hours	0.03827	0.03008	21.41
Volume peak interaction	m^3	0.13475	0.03987	70.40
Density peak interaction	t/m^3	0.00677	0.00138	79.55
Volume step RMSE		0.18983	0.14573	23.23
Density interaction RMSE		0.00132	0.00023	82.65
Volume interaction RMSE		0.02406	0.00720	70.05
Density step RMSE		0.00441	0.00329	25.37

4.5.5 Auto-tuning for improved disturbance rejection using objective function $Q_{BTTreject_2}$

4.5.5.1 Procedure

Closed-loop tests are conducted on the non-linear model of (4.1) by stepping the disturbance. The procedure as discussed in Section 4.4.5.1 applies, except that the plant is now represented by a non-linear model.

4.5.5.2 Simulation

Fig. 4.19 shows the volume and density response for a step disturbance of 0.1 t/m^3 in the Chrome tailings density during the Bayesian optimistation process. The step response of the controller $\boldsymbol{K}_{BTTreject_2}$ with tuning parameters providing the minimum value for objective function $Q_{BTTreject_2}$ is highlighted.

Fig. 4.20 shows how the controller increases the water flow rate to suppress the density disturbance and reduces the tailing supply flow to maintain the sump level at set point. The controlled variables and manipulated variables remain within process bounds for all the candidate tuning parameters applied and evaluated during the optimisation process.

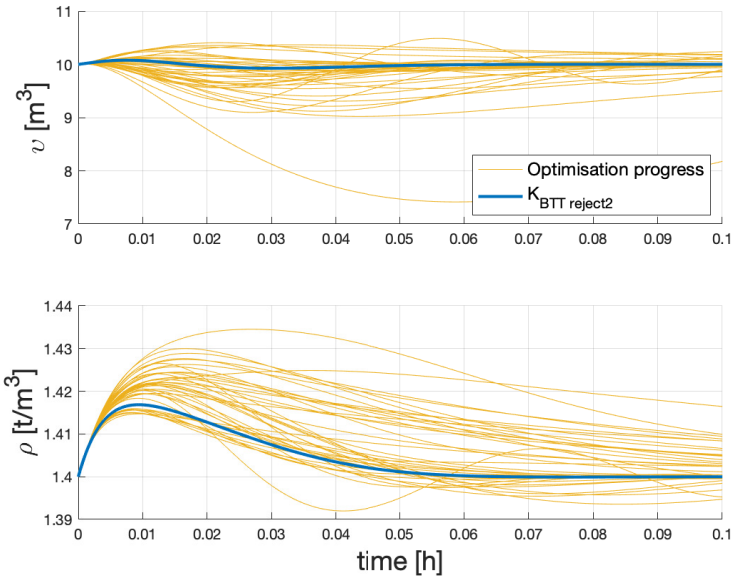

Figure 4.19. Response of the controlled variables to a disturbance step change during Bayesian optimisation using objective function $Q_{BTTreject_2}$.

Table 4.8 shows the value of $Q_{BTTreject_2}$ and associated tuning parameters for iterations results of the Bayesian optimisation simulation using objective function $Q_{BTTreject_2}$, iterations 31 through to 40.

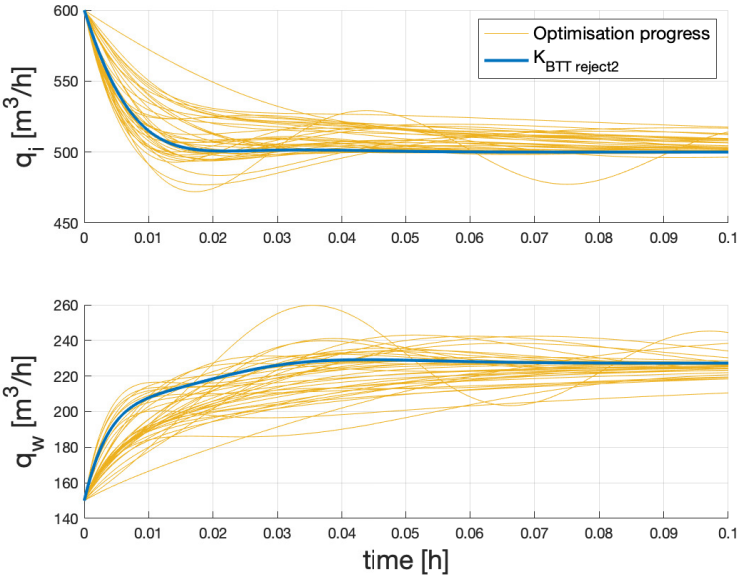

Figure 4.20. Response of the manipulated variables to a disturbance step change during Bayesian optimisation using objective function $Q_{BTTreject_2}$.

The optimisation process requires 34 iterations to show an improvement on controller $\boldsymbol{K}_{inv}$. This is significantly more than the 15 iterations required to minimise the equivalent objective function on the non-linear model. As anticipated, the increased number of iterations is due to the larger number of tuning parameters to be optimised. Objective function $Q_{BTTreject_2}$ only requires a single disturbance step test to evaluate. At 36 minutes per test, the 40 step tests would require no less than 24 hours to complete in practice.

Fig. 4.21 compares the disturbance rejection response of controllers $\boldsymbol{K}_{BTTreject_2}$ and $\boldsymbol{K}_{inv}$. The volume transient time is seen to improve from 0.089 to 0.068 hours, a 24% improvement. The density transient time improves from 0.078 to 0.0581 hours, a 26% improvement.

Table 4.9 summarises the key performance criteria of controller $\boldsymbol{K}_{BTTreject_2}$ and $\boldsymbol{K}_{inv}$. Included in the results are the RMSE calculations confirming that Bayesian optimisation with objective function $Q_{BTTreject_2}$ is successful in auto-tuning $\boldsymbol{K}_{BTTreject_2}$ to improve disturbance rejection performance. These improvements are significant since $\boldsymbol{K}_{inv}$ being an inverse controller has an ideal integrator-type loop with a $-20dB/dec$ roll-off. The results show that an ideal controller can be optimised to promote specified performance criteria captured in a fit for purpose objective function.

Table 4.8. Results of the Bayesian optimisation simulation using objective function $Q_{BTTreject_2}$, iterations 31 through 40.

Iter	$Q_{BTTreject_2}$	k_{P11}	τ_{I11}	k_{P12}	τ_{I12}	k_{P21}	τ_{I21}	k_{P22}	τ_{I22}
31	0.2091	130.42	0.0307	1272	0.0072	37.41	0.0056	-3757	0.0215
32	0.1602	97.701	0.0228	2504	0.0114	37.24	0.0142	-1908	0.0158
33	0.3345	148.22	0.0208	853	0.0098	20.71	0.0057	-1241	0.0423
34	**0.1261**	**156.97**	**0.0144**	**2231**	**0.0092**	**37.01**	**0.0031**	**-2933**	**0.0207**
35	0.1435	133.27	0.0180	3371	0.0130	29.17	0.0032	-1770	0.0116
36	0.1393	141.05	0.0152	1195	0.0113	34.12	0.0038	-3901	0.0154
37	0.3596	93.384	0.0125	1637	0.0133	39.64	0.0167	-2443	0.0382
38	0.1630	143.93	0.0187	3768	0.0092	39.72	0.0051	-2721	0.0324
39	0.1865	106.97	0.0137	2634	0.0082	35.80	0.0117	-3232	0.0261
40	0.4312	92.464	0.0174	3594	0.0281	27.99	0.0127	-3906	0.0501

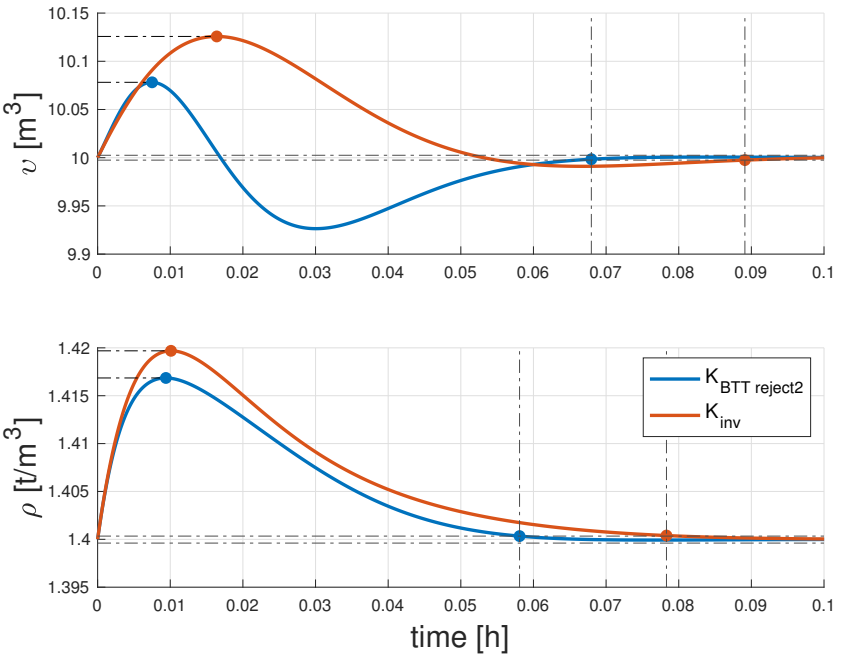

Figure 4.21. Comparison of the disturbance step change response of controllers $\boldsymbol{K}_{BTTreject_2}$ and $\boldsymbol{K}_{inv}$. The markers indicate the improvement of the disturbance peaks and transient times.

Table 4.9. Comparison of disturbance rejection criteria of controller $\boldsymbol{K}_{BTTreject_2}$ and $\boldsymbol{K}_{inv}$.

Performance criteria	Unit	$\boldsymbol{K}_{inv}$	$\boldsymbol{K}_{BTTreject_2}$	Improvement (%)
Volume transient time	hours	0.08910	0.06797	23.72
Density transient time	hours	0.07831	0.05808	25.82
Volume disturbance peak	m^3	0.12577	0.07819	37.83
Density disturbance peak	t/m^3	0.01968	0.01685	14.39
Volume response RMSE		0.00573	0.00169	70.35
Density response RMSE		0.00098	0.00076	22.33

4.6 CONCLUSION

Fit for purpose objective functions are designed to improve set point tracking and disturbance rejection performance respectively.

Settling time is found to be a good performance index to promote set point tracking. The advantage of using of settling time to evaluate the performance of all the controlled variables is that provided the settling time of the controlled variables are similar and the objective function does not implement other performance indices, no scaling factors are required. The disadvantage of an objective function based on settling time is that should the response not settle within the evaluation period, the settling time cannot be measured and the objective function cannot be quantified. Without a scalar objective function value the results cannot be appended to the training dataset $\mathcal{D}$. Settling time can therefore not be used to evaluate unstable controller responses or controllers with a steady state offset greater that 2%.

Transient time is found to be a good performance index to promote disturbance rejection. Similar to settling time, no scaling factor is required provided that the settling time of the controlled variables are similar and the objective function does not consider other performance indices. Transient time suffers from the same disadvantage as settling time.

The constraints of the search domain are based on the assumption that the reference controller is tuned to have a gain margin of two. A gain margin of two is generally accepted to provide a trade-off between performance and robustness (Skogestad and Postlethwaite, 2007). Therefore the reference controller

gains are divided by two and the integral times multiplied by two to approach the instability boundary and set the constraints. The approach may not be successful if the reference controller is already close to a stability margin, in which case the approach proposed may result in unstable iterations. An alternative approach is considered in the next chapter.

The decentralised PI controller has four tuning parameters to optimise, while the inverse controller has double that. It is observed that the optimisation of both set point tracking and disturbance rejection takes longer to converge for the inverse controller than the decentralised PI controller. It is therefore concluded that an increased number of tuning parameters to optimise will require more iterations to converge.

Using RMSE as a common method to compare the performance of the optimised controllers and reference controllers, Bayesian optimisation is shown to improve both the set point tracking as well as disturbance rejection performance of the controllers.

The total optimisation period varies between 12 and 24 hours depending the controller evaluated. This period is the absolute minimum that can be expected, since in practice time would be required to reset after each step test to return and stabilise the process in preparation of the next step. The benefits of the improved performance needs to be weighed against the long evaluation periods, especially if the evaluation periods require production down-time.

For the BBT surge tank process, optimisation of the disturbance rejection controllers is far more effective in terms of the total optimisation time since only a single step test is required. However the simulation is based on the assumption that the disturbance can stepped in practice. If the density of the chrome tailings can not be directly controlled, the disturbance simulations are purely theoretical. Stepping of the controlled variable set points are not subject to the same restrictions and can be changed on demand provided that the operational constraints of Table 4.1 are honoured.

CHAPTER 5 AUTO-TUNING OF ORE MILLING CIRCUIT CONTROLLERS

5.1 CHAPTER OVERVIEW

This chapter presents the approach followed to auto-tune ore milling circuit controllers. The ore milling circuit is considered to be more challenging to control, compared to the BTT surge tank, given the increased dimension of the plant and the stronger interaction between the manipulated and controlled variables.

The objective is to determine if a decentralised PI controller and μ-controller can be optimised using Bayesian optimisation. Considering that the μ-controller does not have tuning parameters, placement of the controller poles are used to optimise the controller.

A novel implementation of robust stability analysis is used to determine the constraints of the search domain.

5.2 ORE MILLING CIRCUIT

The process selected to evaluate the optimisation of the decentralised PI and μ-controllers is the run of mine (ROM) ore milling circuit with three manipulated and three controlled variables. A brief introduction of the process is provided here.

Fig. 5.1 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 discharge grate and collected in the sump where it is diluted with sump feed water. The aperture of the discharge grate determines the particle size distribution of the discharge slurry. The slurry is pumped to a hydrocylone 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.

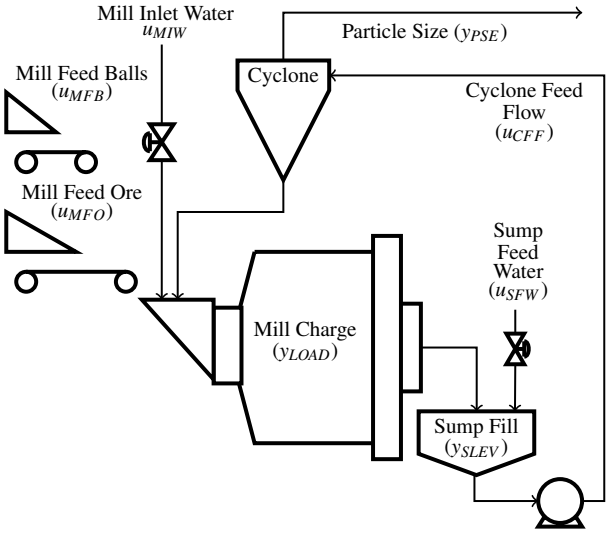

Figure 5.1. ROM ore milling circuit.

Table 5.1 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} (Craig, Hulbert, Metzner and Moult, 1992a), 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 operational points are listed in Table 5.2. At an ore feed rate of $u_{MFO} = 100$ t/h, the mill is charged to $y_{LOAD} = 0.45$ capacity and the sump level is maintained at $y_{SLEV} = 5$ m^3 to provide a product with $y_{PSE} = 0.8$ (Craig and MacLeod, 1995; Coetzee et al., 2010; Le Roux et al., 2013).

Table 5.1. Variable descriptions.

Variable	Description
Controlled variables	
y_{SLEV}	Sump level [m^3]
y_{PSE}	Product particle size estimate [fraction $< 75\ \mu$m]
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 5.2. 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

5.3 AUTO-TUNING OF THE ORE MILLING CIRCUIT DECENTRALISED PI CONTROLLER

5.3.1 Optimisation of set point tracking

Consider the scenario where the milling circuit is controlled by decentralised PI controllers tuned using the SIMC method (Skogestad, 2003) during a desktop study prior to commissioning. The SIMC method requires a linear model to design the controllers. Such a linear model (5.1) is derived from the non-linear model from Coetzee et al. (2010) and Le Roux et al. (2013) and is presented in the form

$$\boldsymbol{y} = \boldsymbol{G}_p(s)\boldsymbol{u} + \boldsymbol{G}_d(s)d \tag{5.1}$$

where

$$\boldsymbol{y} = \left[y_{SLEV}, y_{PSE}, y_{LOAD}\right]^T \tag{5.2}$$

$$\boldsymbol{u} = \left[u_{CFF}, u_{SFW}, u_{MFO}\right]^T. \tag{5.3}$$

The plant transfer function is

$$\boldsymbol{G}_p(s) = \begin{bmatrix} g_{p11} & g_{p12} & g_{p13} \\ g_{p21} & g_{p22} & g_{p23} \\ g_{p31} & g_{p32} & g_{p33} \end{bmatrix} \tag{5.4}$$

where

$$g_{p11} = \frac{-0.29}{s} \tag{5.5a}$$

$$g_{p12} = \frac{0.42}{s} \tag{5.5b}$$

$$g_{p13} = 0 \tag{5.5c}$$

$$g_{p21} = \frac{-0.00035(1-0.63s)}{(1+0.54s)} e^{-0.011s} \tag{5.5d}$$

$$g_{p22} = \frac{0.0055}{1+0.24s} e^{-0.011s} \tag{5.5e}$$

$$g_{p23} = \frac{-0.0043}{1+0.58s} e^{-0.065s} \tag{5.5f}$$

$$g_{p31} = \frac{0.0028(1+0.876s)}{(1+3.868s)} e^{-0.0115s} \tag{5.5g}$$

$$g_{p32} = 0 \tag{5.5h}$$

$$g_{p33} = \frac{0.01}{s}. \tag{5.5i}$$

The disturbance transfer function is

$$\boldsymbol{G}_d(s) = \begin{bmatrix} \dfrac{-0.24}{(1+0.54s)} e^{-0.014s} \\ \dfrac{1.86\times10^{-3}}{(1+14.9s)} e^{-0.438s} \\ \dfrac{0.58}{(1+1.41s)} e^{-0.089s} \end{bmatrix} \tag{5.6}$$

where the disturbance $d = \eta_f$, 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) (Bristol, 1966) 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 k_{ii} on the diagonal.

$$\boldsymbol{K}_\alpha = \begin{bmatrix} k_{11} & 0 & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & k_{33}. \end{bmatrix} \tag{5.7}$$

The PI controllers in Laplace domain are of the form (Seborg et al., 2011)

$$k_{jj} = k_{Pjj}(1 + \frac{1}{\tau_{Ijj}s}), j = 1,2 \tag{5.8}$$

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

$$k_{P11} = -22.9885,\ \tau_{I11} = 0.6 \tag{5.9a}$$

$$k_{P22} = 206.8074,\ \tau_{I22} = 0.24 \tag{5.9b}$$

$$k_{P33} = 500,\ \tau_{I33} = 0.8. \tag{5.9c}$$

Consider that post commissioning, as part of production performance evaluation, closed-loop set point step tests are conducted on the full model of (5.4) using the controller $\boldsymbol{K}_\alpha$ defined by (5.7) through (5.9) to gauge the performance. The results captured in Figs. 5.7 and 5.8 show that the response of y_{PSE} is overdamped with a settling time of 1.47 hours. The response of y_{LOAD} is underdamped with a peak overshoot of approximately 14% and a settling time of 2.09 hours. The performance may be considered inadequate when plant-model mismatch, equipment replacement or deterioration of equipment performance are taken into account. Improving the set point tracking ability of the controller is required to benefit the supervisory layer of a production or economic optimiser (Craig, Hulbert, Metzner and Moult, 1992b; Le Roux and Craig, 2019). To this end, Bayesian Optimisation is applied as a model free, 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.

5.3.1.1 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 minimize 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 (Skogestad and Postlethwaite, 2007; MATLAB, 2022) 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 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

$$k_{P11} \in [-52.69, -0.539] \tag{5.10a}$$

$$\tau_{I11} \in [0.262, 25.55] \tag{5.10b}$$

$$k_{P22} \in [4.413, 473.2] \tag{5.10c}$$

$$\tau_{I22} \in [0.105, 10.22] \tag{5.10d}$$

$$k_{P33} \in [11.71, 1146] \tag{5.10e}$$

$$\tau_{I33} \in [0.349, 34.07]. \tag{5.10f}$$

By expanding the search domain to the threshold of instability as given in (5.10), the probability of including the optimal tuning parameters to find the global minimum of the objective function is increased.

Fig. 5.2 shows the robust stability μ plot with the tuning parameter search domain constrained as per (5.10). 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 (5.10), will remain stable during Bayesian optimisation. In addition, the

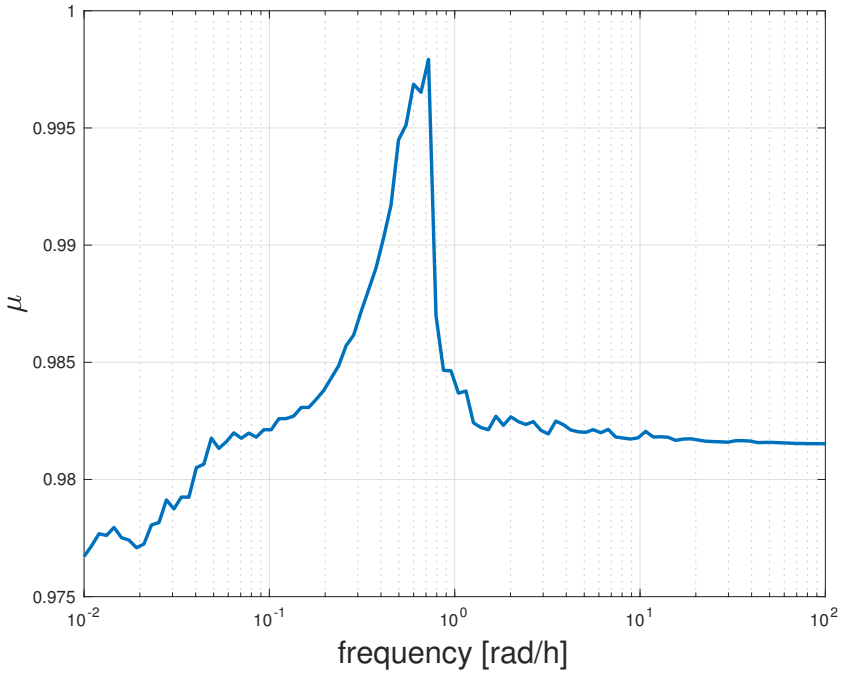

Figure 5.2. Robust stability structured singular value (μ) plot with tuning parameters constrained as per (5.10).

μ values are just below 1 over the frequency range of interest, as they should be given that parameter ranges of (5.10) represent the closed-loop system at the threshold of instability.

5.3.2 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 (3.3), is

$$Q_{track} = \omega_1 \beta_{11} q_1 + \omega_2 \beta_{22} q_2. \tag{5.11}$$

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 the 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. 5.7 and 5.8 show that the transient dynamics of $\boldsymbol{K}_\alpha$ have mostly decayed after 2 hours 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 toward 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 hours to settle. Evaluation periods of 20 hours per step tests are impractical if step tests result sub-optimal process performance 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 (5.11) are $\beta_{11} = \frac{1}{0.9137}$ and $\beta_{22} = \frac{1}{0.7498}$. The scaling factors are the inverse ITAE values in response to set point step changes of controller $\boldsymbol{K}_\alpha$ integrated over a 2 hour period which requires a step test to calculate.

5.3.3 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 (Galán, Barton and Romagnoli, 2002). Karageorgos, Genovese and

Baas (2006) 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 (Galán et al., 2002).

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.

5.3.4 Objective function for disturbance rejection

The objective function selected to retune controller $\boldsymbol{K}_\alpha$ for improved disturbance rejection, presented in the form of (3.3), is

$$Q_{reject} = \omega_1\beta_{11}q_1 + \omega_2\beta_{22}q_2 + \omega_3\beta_{33}q_3. \tag{5.12}$$

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 (5.12) 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 hour period.

5.4 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.

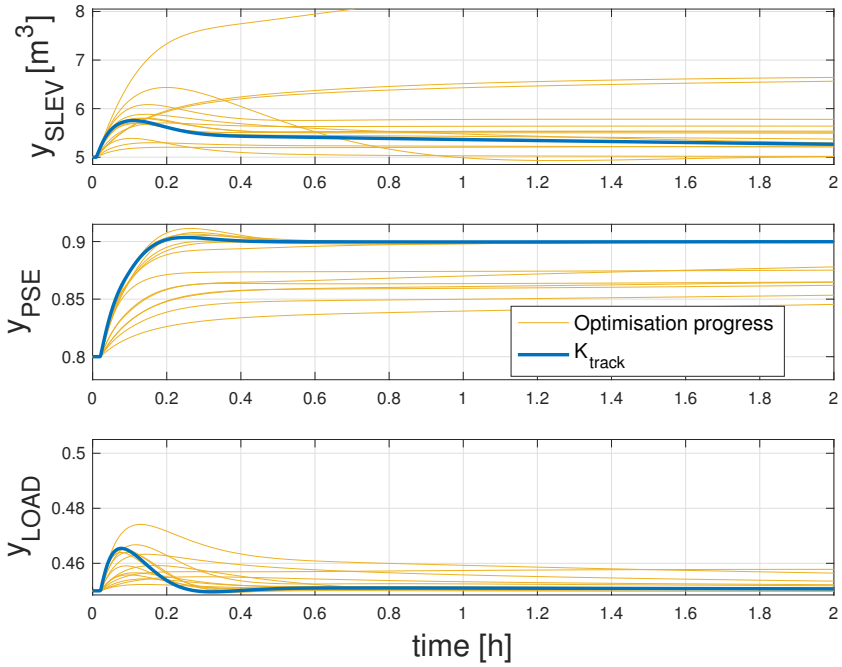

Figure 5.3. Response of the controlled variables to a y_{PSE} set point step change during Bayesian optimisation using objective function Q_{track}.

5.4.1 Set point tracking

Figs. 5.3 and 5.4 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 (5.7) by minimising Q_{track} in (5.11). 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. 5.5 and 5.6 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

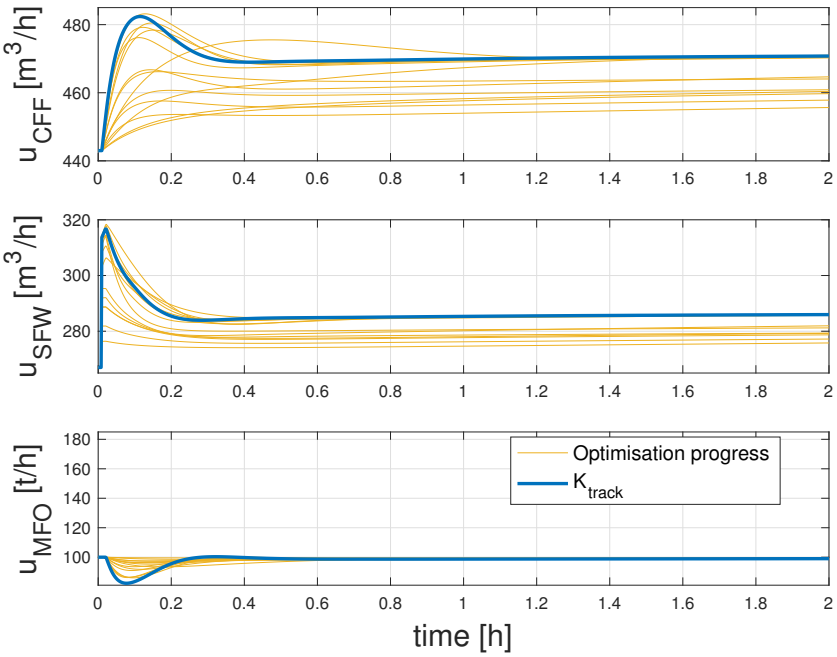

Figure 5.4. Response of the manipulated variables to a y_{PSE} set point step change during Bayesian optimisation using objective function Q_{track}.

y_{LOAD}. y_{PSE} decreases due to the increased u_{MFO} and as a result the u_{SFW} increases to return y_{PSE} to the set point. 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. 5.7 and 5.8 show the response of y_{PSE} and y_{LOAD} to step changes and compares the tracking performance of controller $\boldsymbol{K}_{\alpha}$ and the controller retuned using Bayesian optimisation $\boldsymbol{K}_{track}$. Objective function (5.11), selected to improve set point tracking, can be seen to improve the y_{PSE} settling time from 1.47 to 0.32 hours. The y_{LOAD} settling time is reduced from 2.09 to 0.22 hours and the peak amplitude reduced from a fraction of 0.507 to 0.5. Table 5.3 lists the ITAE value reduction which is the basis of objective function (5.11). 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 hour period.

Table 5.4 shows the results of iterations 6 through to 15 of the Bayesian optimisation simulation using objective function (5.11). 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 hours.

Table 5.3. 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. (%)
$y_{PSE\ ITAE}$	0.2097	0.9137	77.1
$y_{LOAD\ ITAE}$	0.1225	0.7498	83.7
$y_{PSE\ RMSE}$	0.0302	0.044	31.5
$y_{LOAD\ RMSE}$	0.0153	0.0213	28.0
$y_{PSE\ SettlingTime}$	0.32	1.47	86.7
$y_{LOAD\ SettlingTime}$	0.22	2.09	89.5

Table 5.4. Results of Bayesian optimisation simulation using objective function (5.11), iterations 6 through 15.

Iteration	Q_{track}
6	7.7665
7	0.88053
8	0.51078
9	6.7141
10	0.66516
11	2.1642
12	0.49216
13	**0.39279**
14	1.0471
15	0.55844

With each Bayesian iteration requiring two step tests, the 15 iterations as suggested in Table 5.4 will require no less than 60 hours to complete in practice. The best result is found by iteration 13. The results achieved as shown in Figs. 5.7 and 5.8 are satisfactory and conducting further iterations in search of the global minimum at an overhead of 4 hours per iteration does not warrant any further increase in performance.

The tuning parameters corresponding to the best iteration are

$$k_{P11} = -50.772,\ \tau_{I11} = 2.7411 \tag{5.13a}$$

$$k_{P22} = 466.81,\ \tau_{I22} = 0.15388 \tag{5.13b}$$

$$k_{P33} = 1144.2,\ \tau_{I33} = 21.105. \tag{5.13c}$$

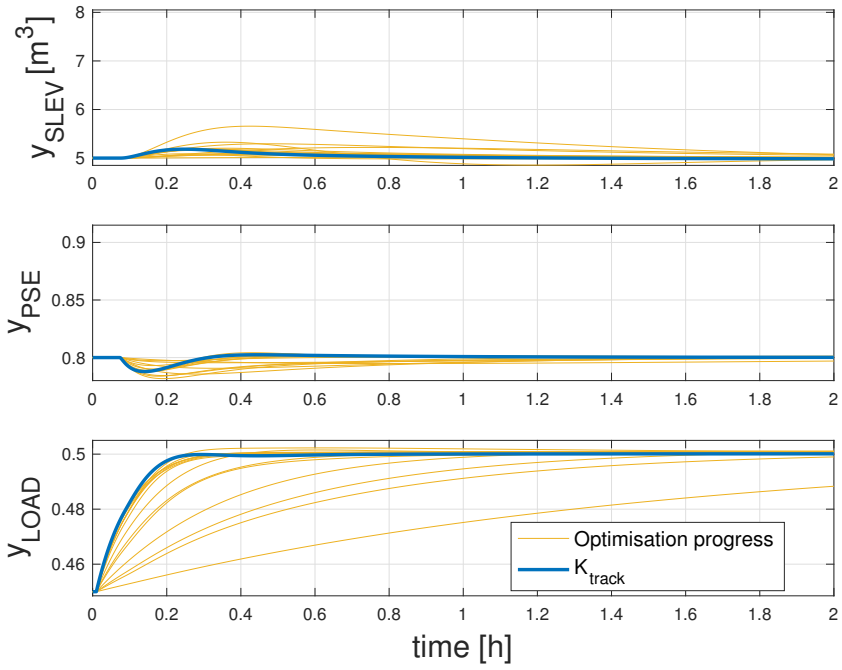

Figure 5.5. Response of the controlled variables to a y_{LOAD} set point step change during Bayesian optimisation using objective function Q_{track}.

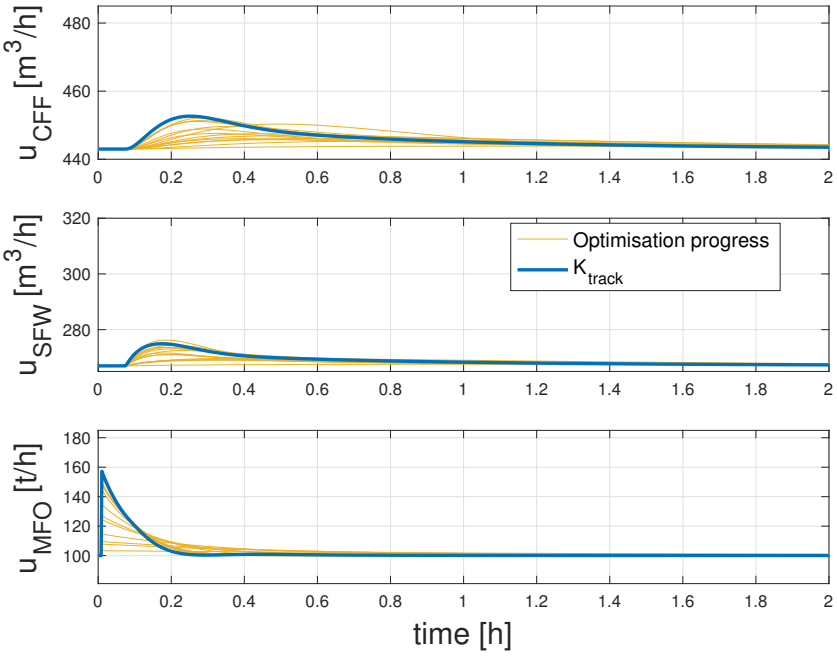

Figure 5.6. Response of the manipulated variables to a y_{LOAD} set point step change during Bayesian optimisation using objective function Q_{track}.

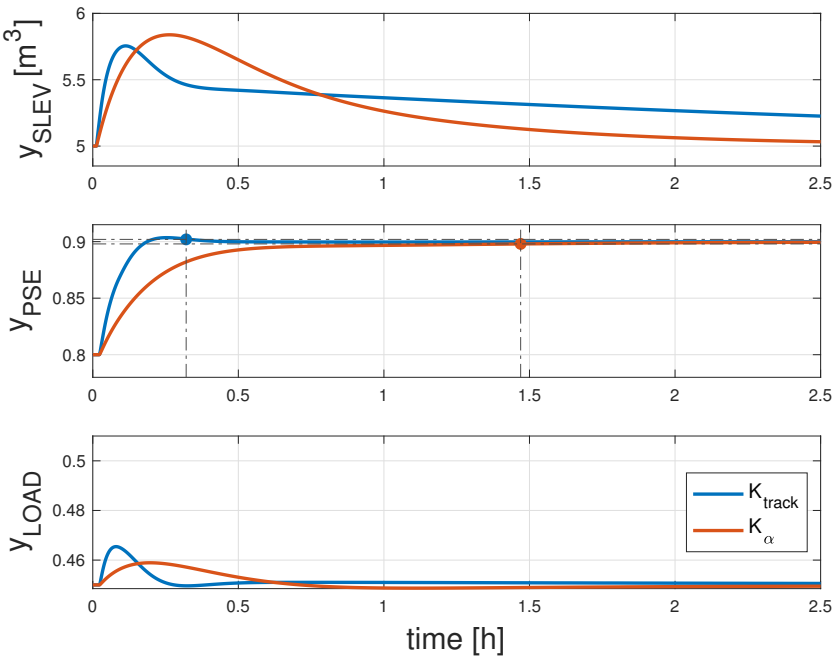

Figure 5.7. Comparison of the set point tracking performance of controllers $\boldsymbol{K}_{track}$ and $\boldsymbol{K}_{\alpha}$ in response to a y_{PSE} set point step change. The markers indicate the settling time of the responses.

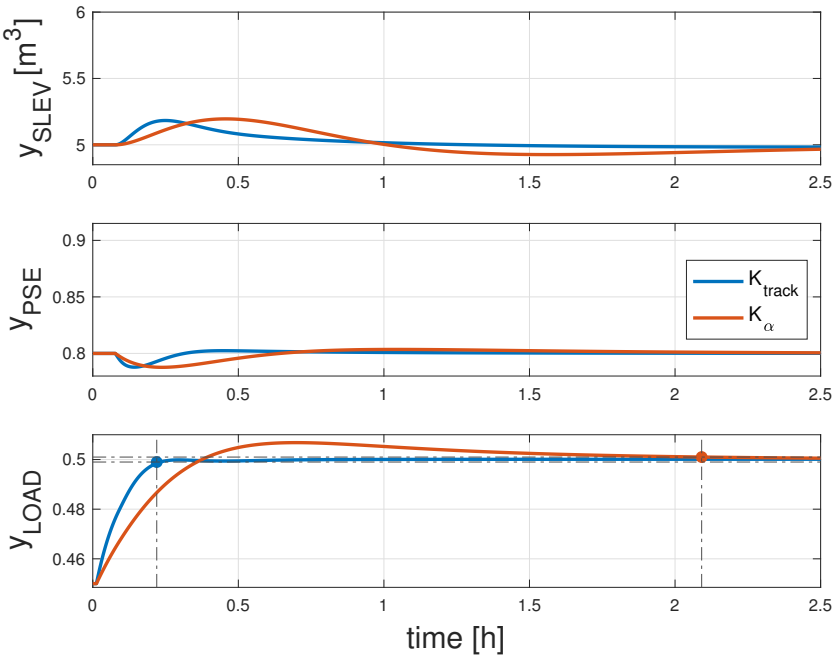

Figure 5.8. Comparison of the set point tracking performance of controllers $\boldsymbol{K}_{track}$ and $\boldsymbol{K}_{\alpha}$ in response to a y_{LOAD} set point step change. The markers indicate the settling time of the responses.

.

5.4.2 Disturbance rejection

Figs. 5.9 and 5.10 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 (5.7) by minimising Q_{reject} in (5.12). 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. Should this occur in practice, the sump slurry pump will trip due to the low sump level, and the optimisation interation will abort. u_{CFF} is close to the maximum limit indicating that the plant and decentralised PI controller will not be able to cater for ore hardness disturbances much greater than 2.5% before u_{CFF} saturates and the sump overflows.

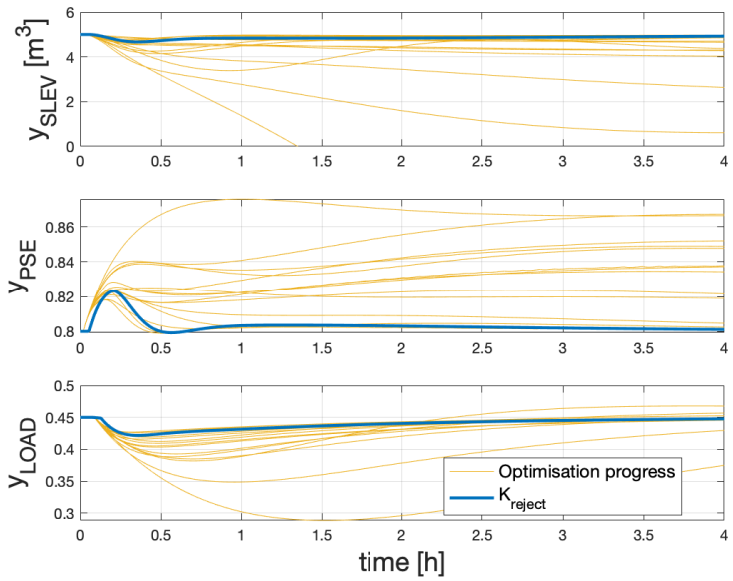

Figure 5.9. Response of the controlled variables to an ore hardness step change during Bayesian optimisation using objective function Q_{reject}.

Fig. 5.11 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 5.5 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}

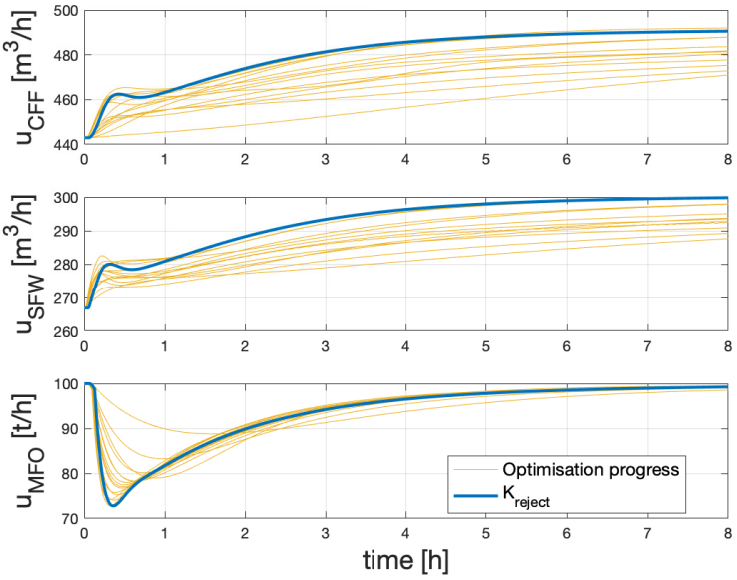

Figure 5.10. Response of the manipulated variables to an ore hardness step change during Bayesian optimisation using objective function Q_{reject}.

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 hour 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 5.5 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. 5.11 it is evident that the transients due to disturbances take much longer to decay compared to the set point step changes of Figs. 5.7 and 5.8. 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 hours 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

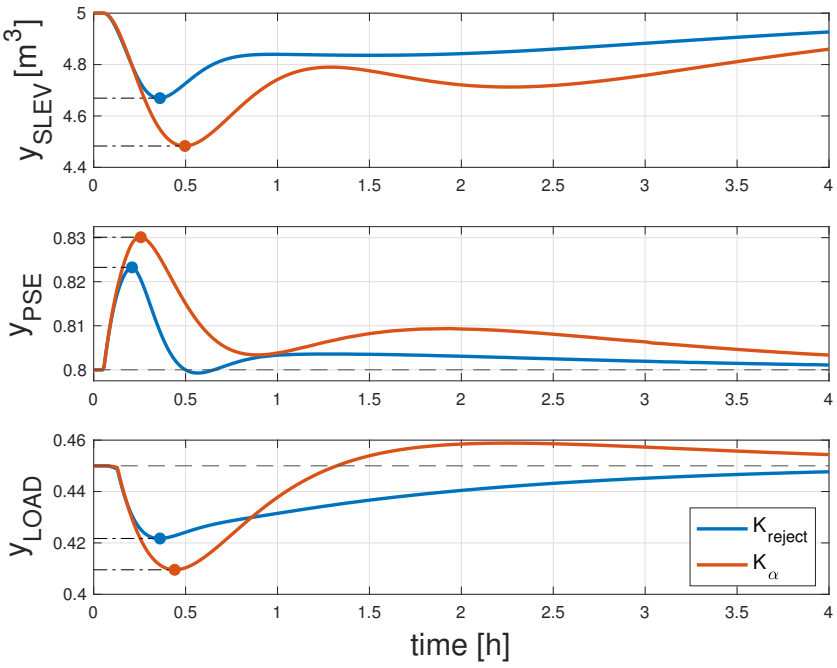

Figure 5.11. Comparison of the disturbance rejection performance of controllers $\boldsymbol{K}_{reject}$ and $\boldsymbol{K}_{\alpha}$ in response to an ore hardness step change. The markers indicate the peak disturbance error of the responses.

Table 5.5. 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
$y_{SLEV\ RMSE}$	0.1151	0.2001	42.5
$y_{PSE\ RMSE}$	0.0038	0.0073	47.1
$y_{LOAD\ RMSE}$	0.0094	0.0108	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

response to a set point step change.

Simulations show that evaluation periods of up to 24 hours are required for the objective function to provide a useful training dataset $\mathcal{D}$ making transient time an unsuitable performance index for disturbance rejection.

Table 5.6 shows the results of iterations 6 through to 15 of the Bayesian optimisation simulation using objective function (5.12). During simulation, the step test response was evaluated over a period of 4 hours to calculate the ITAE. From Fig. 5.11 it can be seen that the disturbance peaks have decayed after 4 hours. With each Bayesian optimisation iteration only requiring a single step, the 15 iterations as suggested in Table 5.6 would require no less than 60 hours 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.

Table 5.6. Results of Bayesian optimisation simulation using objective function (5.12), iterations 6 through 15.

Iteration	Q_{reject}
6	0.048967
7	0.087028
8	0.035282
9	0.13597
10	0.022599
11	0.056741
12	**0.013134**
13	0.017427
14	0.43977
15	0.084509

The tuning parameters corresponding to the best iteration are

$$k_{P11} = -45.607,\ \tau_{I11} = 0.68394 \tag{5.14a}$$

$$k_{P22} = 247.97,\ \tau_{I22} = 0.11517 \tag{5.14b}$$

$$k_{P33} = 956.8,\ \tau_{I33} = 32.947. \tag{5.14c}$$

5.5 AUTO-TUNING OF THE ORE MILLING CIRCUIT μ-CONTROLLER

5.5.1 Plant Model

Bayesian optimisation has been shown to successfully auto-tune decentralised PI and inverse controllers for improved performance. The controllers auto-tuned thus far all had PI controllers embedded on the elements in the controller matrix. This section investigates if the more complex μ-controller structure can benefit from Bayesian optimisation. A μ-controller consist of multi-order polynomial transfer functions on all the elements within the controller matrix structure.

Figure 5.1 represents ore milling process, but a different plant model is selected to construct the μ-controller. The transfer function model (5.15) as presented in the appendix of Craig and MacLeod (1996) is used. This model is selected since the uncertainty descriptions, uncertainty weights and performance weights have been selected and motivated by Craig and MacLeod (1996). It is not the focus of this research to argue the uncertainty descriptions, uncertainty weights and performance weights to synthesise the μ-controller but rather to investigate if Bayesian optimisation can improve a μ-controller's performance parameter of choice.

The transfer function model of the ore milling circuit in the form of $\boldsymbol{y} = \boldsymbol{G}(s)\boldsymbol{u}$, determined from multiple step tests conducted by Craig and MacLeod (1996) is

$$\begin{bmatrix} y_{PSE} \\ y_{LOAD} \\ y_{SLEV} \end{bmatrix} = \begin{bmatrix} \frac{0.14}{175s+1}e^{-40s} & \frac{-0.082}{1766s+1}e^{-620s} & \frac{-0.0575}{167s+1}e^{-40s} \\ 0 & \frac{2.21\times10^{-5}}{s} & 0 \\ \frac{0.00253}{s} & 0 & \frac{-0.00299}{s} \end{bmatrix} \begin{bmatrix} u_{SFW} \\ u_{MFS} \\ u_{CFF} \end{bmatrix}. \tag{5.15}$$

5.5.2 Controller

Using the uncertainty descriptions, uncertainty weights, performance weights, and control weights from Craig and MacLeod (1996), a μ-controller is synthesised using the Matlab robust control toolbox. To reduce the computational effort of the implemented controller and remain within the dimensional limits of Bayesian optimisation, the order of the μ-controller, referred to as $\boldsymbol{K}_{\mu 67}$, is reduced from 67 to 10, the lowest order at which the robust performance μ-curve peak is < 1.0. By keeping the robust performance μ-curve < 1.0 the closed-loop performance of the controller will meet the specified

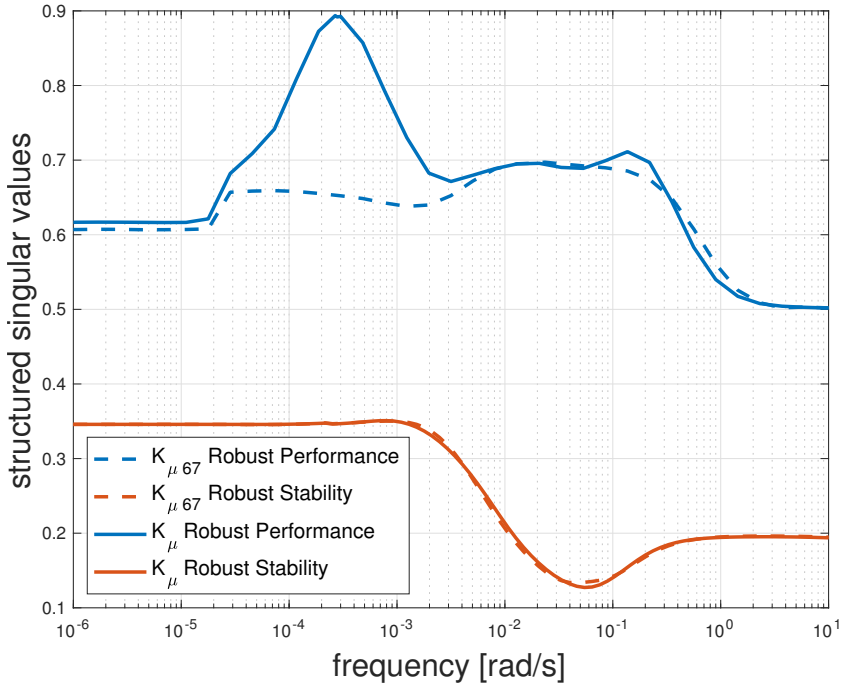

Figure 5.12. Robust stability and robust performance structured singular value plots of the $\boldsymbol{K}_{\mu 67}$ and $\boldsymbol{K}_{\mu}$ controller. The plots of $\boldsymbol{K}_{\mu 67}$ are trended as dashed lines, while the plots of $\boldsymbol{K}_{\mu}$ are solid lines.

performance criteria given the uncertain process model. The reduced order μ-controller in transfer function form is represented by

$$\boldsymbol{K}_{\mu} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \tag{5.16}$$

where each matrix entry has the form

$$k_{ij} = \frac{\beta_{ij12}s^9 + \cdots + \beta_{ij1}s + \beta_{ij0}}{s^{10} + a_{ij12}s^9 \cdots + a_{ij1}s + a_{ij0}}, i, j = 1, 2, 3. \tag{5.17}$$

Fig. 5.12 compares the structured singular values μ for robust performance and robust stability of $\boldsymbol{K}_{\mu 67}$ and reduced order controller $\boldsymbol{K}_{\mu}$ of (5.16). As a result of the model reduction, it can be seen that the robust performance deteriorates from a peak μ of 0.69 to 0.89, while the robust stability remains essentially unchanged . $\boldsymbol{K}_{\mu}$ will therefore remain robustly stable but will not be able to achieve the same level of performance as $\boldsymbol{K}_{\mu 67}$ for the modelled uncertainty.

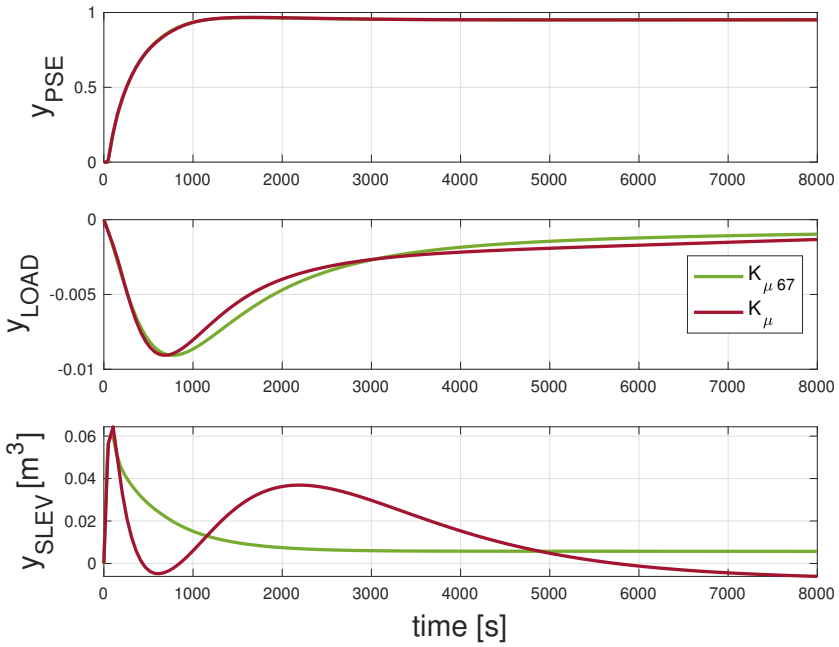

Figure 5.13. Comparison of the set point tracking performance of controllers $\boldsymbol{K}_{\mu 67}$ and $\boldsymbol{K}_{\mu}$ in response to a y_{PSE} set point step change.

Figs. 5.13 and 5.14 compare the responses of controller $\boldsymbol{K}_{\mu 67}$ and the reduced order controller K_{μ} to y_{PSE} and y_{LOAD} set point step changes, to determine if the reduced model has degraded the set point tracking performance. Note that the plant has been scaled for controller synthesis and Figs. 5.13 and 5.14 present the scaled controlled variables.

From the y_{PSE} set point step change response in Fig. 5.13, it can be seen that the responses of y_{PSE} and y_{LOAD} of $\boldsymbol{K}_{\mu}$ compare well to that of $\boldsymbol{K}_{\mu 67}$, but $\boldsymbol{K}_{\mu}$ is not able to suppress the interaction with y_{SLEV} as successfully as $\boldsymbol{K}_{\mu 67}$.

From the y_{LOAD} set point step change response in Fig. 5.14 it can be seen that the responses of y_{PSE} and y_{LOAD} of $\boldsymbol{K}_{\mu}$ compare well to that of $\boldsymbol{K}_{\mu 67}$. Once again controller $\boldsymbol{K}_{\mu}$ does not manage to suppress the interaction with y_{SLEV} as well as $\boldsymbol{K}_{\mu 67}$.

Set point tracking of y_{SLEV} has no economic benefit and is not considered. The interaction with y_{SLEV} can be tolerated as long as the sump does not run dry or overflow.

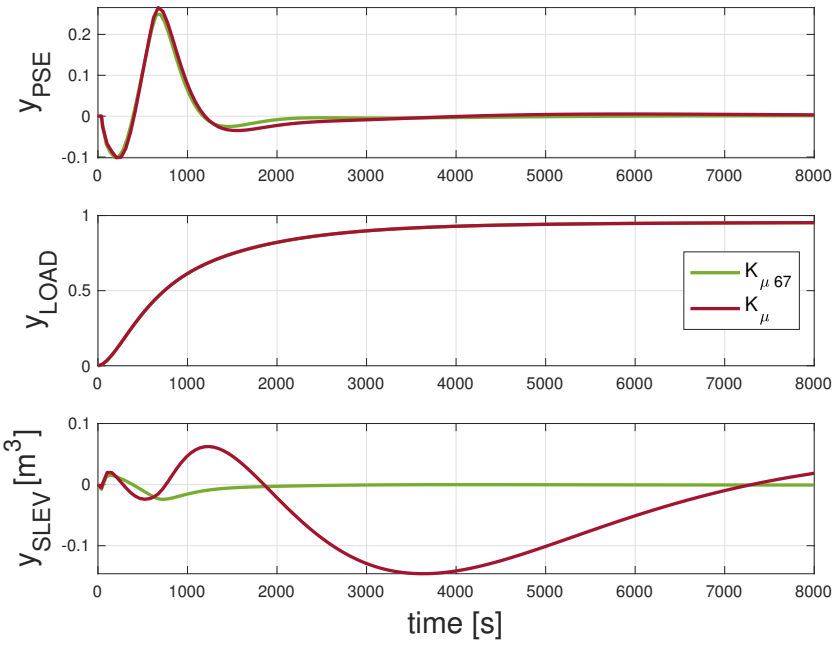

Figure 5.14. Comparison of the set point tracking performance of controllers $\boldsymbol{K}_{\mu 67}$ and $\boldsymbol{K}_{\mu}$ in response to a y_{LOAD} set point step change.

5.5.3 Optimisation of set point tracking

To optimise the performance of controller $\boldsymbol{K}_{\mu}$ for set point tracking, $\boldsymbol{K}_{\mu}$ is converted from the transfer function form of (5.16) and (5.17) to the state space form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{e}(t) \tag{5.18a}$$

$$\boldsymbol{u}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{e}(t) \tag{5.18b}$$

where $\boldsymbol{x}(t)$ is the controller state vector, $\boldsymbol{u}(t)$ is the controller output or manipulated variable vector, $\boldsymbol{e}(t)$ is the control error vector which is the difference between the set points and the controlled variables. Matrix $\boldsymbol{A}$ is the state matrix, $\boldsymbol{B}$ is the input matrix, $\boldsymbol{C}$ is the output matrix and $\boldsymbol{D}$ is the feed through matrix.

The stability of the controller is solely determined by $\boldsymbol{A}$, the $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ matrices have no effect (Seborg et al., 2011) and is therefore selected for optimisation to improve the set point tracking of $\boldsymbol{K}_{\mu}$.

The order of $\boldsymbol{K}_{\mu}$ has been reduced to 10, therefore $\boldsymbol{A}$ is a 10-by-10 matrix and could contain up to a 100 element to optimise. Given the practical limitations of Bayesian optimisation to process a large number of parameters (Moriconi, Deisenroth and Sesh Kumar, 2020), $\boldsymbol{A}$ is diagonalised to reduce the

number of matrix elements to 10.

The eigenvalues of a diagonal matrix is equal to the diagonal elements of the matrix (Skogestad and Postlethwaite, 2007). The controller is stable if the eigenvalues of the matrix $\boldsymbol{A}$ have negative real parts, that is, lie in the open left-half plane (Skogestad and Postlethwaite, 2007). The eigenvalues of $\boldsymbol{A}$ are equivalent to the poles of the transfer function and therefore adjusting the eigenvalues of $\boldsymbol{A}$ is equivalent to changing the positions of the controller poles. The controller $\boldsymbol{K}_\mu$ will be optimised by identifying the optimal pole positions for improved set point tracking.

5.5.4 Constraints

The diagonal matrix elements of the state matrix $\boldsymbol{A}$, and poles of controller $\boldsymbol{K}_\mu$, are

$$\boldsymbol{A} = diag(a_1, a_2, \ldots, a_{10}). \tag{5.19}$$

where

$$a_1 = -0.286 \tag{5.20a}$$

$$a_2 = -0.0474 \tag{5.20b}$$

$$a_3 = -0.0394 \tag{5.20c}$$

$$a_4 = -0.00543 \tag{5.20d}$$

$$a_5 = -0.00382 \tag{5.20e}$$

$$a_6 = -0.00071 \tag{5.20f}$$

$$a_7 = -0.00025 \tag{5.20g}$$

$$a_8 = -0.000115 \tag{5.20h}$$

$$a_9 = -5.31 \times 10^{-05} \tag{5.20i}$$

$$a_{10} = -2.75 \times 10^{-05}. \tag{5.20j}$$

The constraint must be selected large enough to include optimal pole positions, but small enough to exclude unstable pole positions. Unlike the PI tuning parameters where intuition can guide the selection of constraints, there is no intuitive approach for constraining the search domain of the poles, apart from keeping them in the left-half plane. To expand the search space around the poles of the controller $\boldsymbol{K}_\mu$, a robust stability analysis (Skogestad and Postlethwaite, 2007; MATLAB, 2022) is conducted on an initial set of constraints to determine how much uncertainty over and above the initial constraints can be tolerated. The initial constraints are simply selected as $\pm 5\%$ of the pole value (5.20).

The Robust Control Toolbox of MATLAB provides the stability margins for the uncertain system incorporating $\boldsymbol{K}_\mu$ with uncertain pole positions. The robust stability analysis provides the maximum pole position uncertainty that can be tolerated before the worst-case uncertainty yields instability. The maximum pole position uncertainty determines the constraints of the search domain which are

$$a_1 \in [-0.301, -0.272] \tag{5.21a}$$

$$a_2 \in [-0.0497, -0.045] \tag{5.21b}$$

$$a_3 \in [-0.0413, -0.0374] \tag{5.21c}$$

$$a_4 \in [-0.0057, -0.00516] \tag{5.21d}$$

$$a_5 \in [-0.00401, -0.00363] \tag{5.21e}$$

$$a_6 \in [-0.000746, -0.000675] \tag{5.21f}$$

$$a_7 \in [-0.000263, -0.000238] \tag{5.21g}$$

$$a_8 \in [-0.000121, -0.000109] \tag{5.21h}$$

$$a_9 \in [-5.57 \times 10^{-05}, -5.04 \times 10^{-05}] \tag{5.21i}$$

$$a_10 \in [-2.88 \times 10^{-05}, -2.61 \times 10^{-05}]. \tag{5.21j}$$

By expanding the search domain to the threshold of instability as given in (5.21), the probability of including the optimal pole positions to find the global minimum of the objective function is increased.

Fig. (5.15) shows the robust stability μ plot with the pole position search domain constrained as per (5.21). Since $\mu < 1$ for all frequencies, it confirms that the closed-loop transfer function of $\boldsymbol{G}$ and $\boldsymbol{K}_\mu$ will remain stable during Bayesian optimisation.

5.5.5 Objective function

The objective of the optimisation is to improve the set point tracking performance of the controller $\boldsymbol{K}_\mu$. Improving the set point tracking ability of the controller could be required to benefit the supervisory layer of a production or economic optimiser (Craig et al., 1992b). To meet the auto-tuning objective, the poles of $\boldsymbol{K}_\mu$ must be optimally placed such that the settling time is reduced after a set point change.

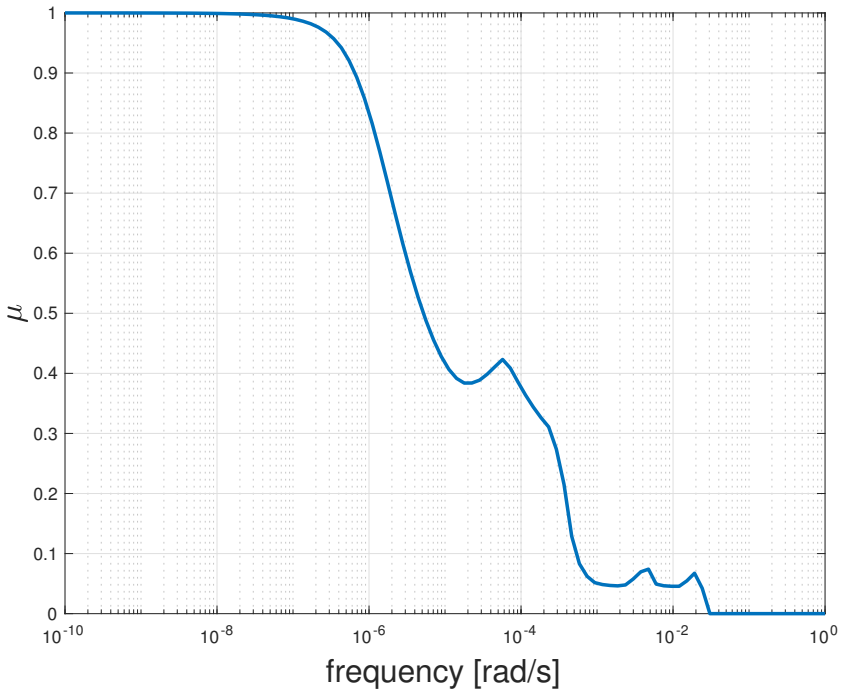

Figure 5.15. Robust stability structured singular value (μ) plot with poles constrained as per (5.21).

The purpose of objective function $Q_{\mu track}$ is to auto-tune controller $\boldsymbol{K}_{\mu}$ for improved set point tracking. Objective function $Q_{\mu track}$ in the form of (3.3) is

$$Q_{\mu track} = \omega_1 \beta_{11} q_1 + \omega_2 \beta_{22} q_2. \tag{5.22}$$

Objective function $Q_{\mu track}$ consist of two terms. The first term represents the tracking performance of y_{PSE} and the second term that of y_{LOAD}. Each term requires a step test to evaluate, therefore each Bayesian optimisation iteration will require two step tests. Both outputs are considered to be of equal importance therefore $\omega_1 = \omega_1 = 1$.

The performance index q_1 is the ITAE of y_{PSE} and q_2 is the ITAE of y_{LOAD}. Using the ITAE criteria as performance index has the benefit that the objective function value can be calculated regardless of whether or not the response settles during the evaluation period. From Figs. 5.13 and 5.14 it can be seen that after 8000 seconds (2.2 hours), the set point responses and interactions have mostly settled and is therefore selected as the evaluation period.

During simulation it was found that using settling time as a performance index was not feasible. With an evaluation period of 4.4 hours, as many as 20 consecutive iterations did not settle within the evaluation

time. As a result, none of these iterations contributed to the training dataset $\mathcal{D}$. Increasing the evaluation period is an option but was discarded as this would significantly increase the optimisation duration which is unpractical given that alternative performance criteria such as ITAE are available.

The ITAE values of y_{PSE} and y_{LOAD} are significantly different, and must be scaled to ensure that they contribute equally to the objective function value. The scaling factors used in (5.22) are $\beta_{11} = \frac{1}{29717}$ and $\beta_{22} = \frac{1}{48760}$. These scaling factors are the inverse ITAE values in response to set point step changes of $\boldsymbol{K}_{\mu}$ integrated over a period of 8000 seconds.

5.5.6 Auto-tuning for improved set point tracking using $Q_{\mu track}$

5.5.6.1 Procedure

The procedure followed is suitable for simulation, but to implement the procedure in practice will require the unscaling of the controller.

Closed-loop step tests are conducted on the scaled linear plant by stepping the set points of y_{PSE} and y_{LOAD} with a unitary value. Two step tests are required for each Bayesian optimisation iteration because the objective function evaluates both the y_{PSE} and y_{LOAD} outputs. The acquisition function will adjust the positions of the poles before each iteration with the objective of minimising the objective function value.

5.5.6.2 Simulation

Figs. 5.16 and 5.17 show the results of the set point step changes and the interaction with the non-stepped outputs. The figures show how Bayesian optimisation explores different pole positions with the intention of minimising the objective function $Q_{\mu track}$. The step response of the best performing controller $\boldsymbol{K}_{\mu track}$ is highlighted. Fig. 5.17 shows that during one the iterations, y_{SLEV} deviates by more that 50% from set point indicating that the sump will overflow if the sump set point is at the nominal set point of 50%.

Figs. 5.18 and 5.19 compares the set point step response of controllers $\boldsymbol{K}_{\mu}$ and $\boldsymbol{K}_{\mu track}$. Controller $\boldsymbol{K}_{\mu}$ is the reduced order controller and $\boldsymbol{K}_{\mu track}$ is the controller optimised for improved set point tracking using Bayesian optimisation.

Fig. 5.18 shows the response to a y_{PSE} set point step change. There is a slight improvement in the y_{PSE} settling time from 1017 to 993 seconds, which is a 2% improvement. The minor improvement comes

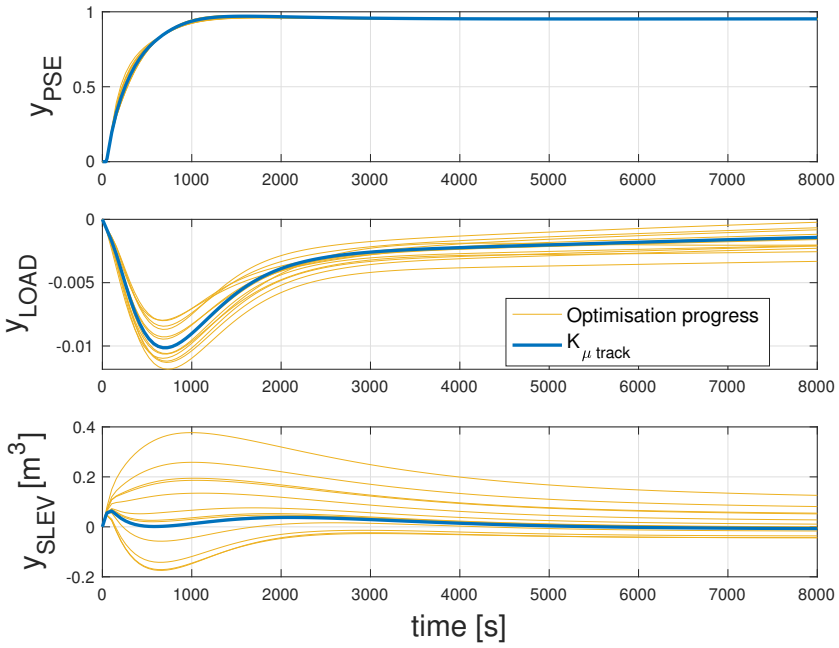

Figure 5.16. Response of the controlled variables to a y_{PSE} set point step change during Bayesian optimisation using objective function $Q_{\mu track}$.

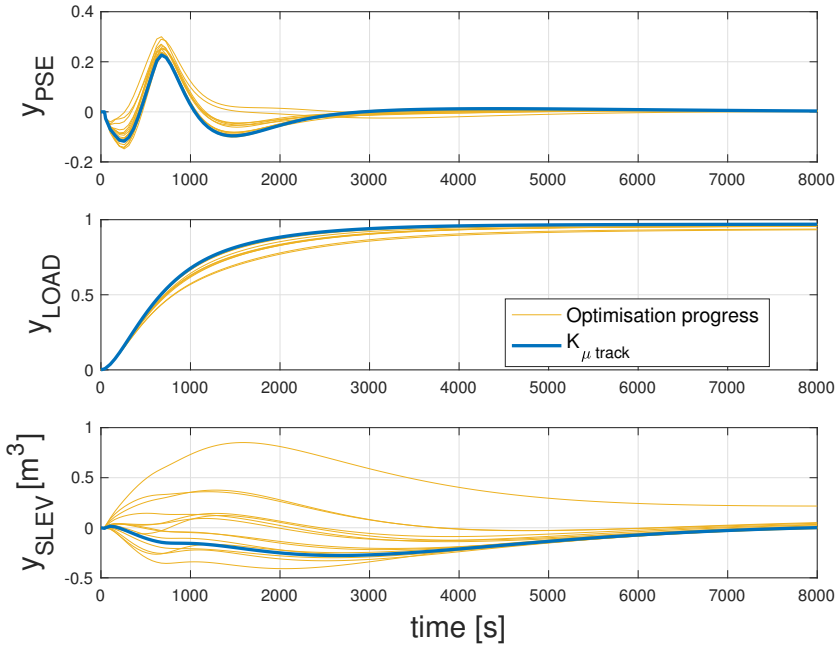

Figure 5.17. Response of the controlled variables to a y_{LOAD} set point step change during Bayesian optimisation using objective function $Q_{\mu track}$.

at the expense of a larger interaction peak with y_{LOAD}. The interaction with y_{SLEV} is negligible as the sump will not run dry or overflow if the sump set point is maintained at 50%.

Fig. 5.19 shows the response to a y_{LOAD} set point step change. There is a significant improvement in the y_{LOAD} settling time from 61478 to 27234 seconds, which is a 56% improvement. The interaction peak with y_{PSE} is also slightly improved. The improved y_{LOAD} set point tracking comes at the expense of significant interaction with y_{SLEV}.

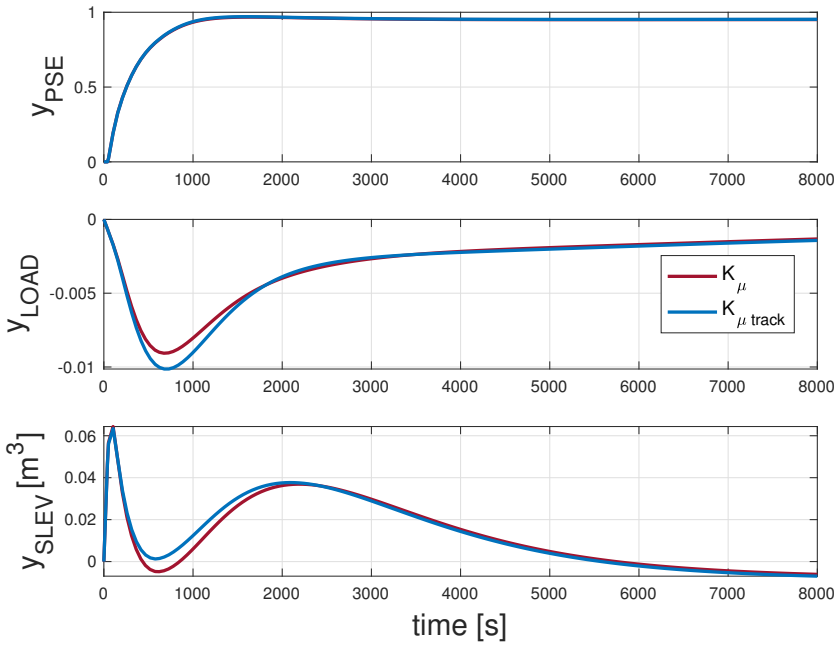

Figure 5.18. Comparison of the set point tracking performance of controllers $\boldsymbol{K}_{\mu}$ and $\boldsymbol{K}_{\mu track}$ in response to a y_{PSE} set point step change.

To determine how the optimisation has influenced the robust performance and stability of the controller, Fig. 5.20 compares the structured singular value plots of $\boldsymbol{K}_{\mu}$ to that of $\boldsymbol{K}_{\mu track}$. The robust stability of the two controllers is similar but the robust performance of $\boldsymbol{K}_{\mu track}$ deteriorates from 0.89 to 1.1 and therefore no longer meets the performance requirements for all the modelled uncertainties.

Table 5.7 summarises the key performance criteria of controllers $\boldsymbol{K}_{\mu}$ and $\boldsymbol{K}_{\mu track}$. Included in the results are the RMSE calculations. The results confirm that y_{LOAD} set point tracking has improved, but there is only minimal improvement on the set point tracking of y_{PSE}. Even though Bayesian optimisation does not show much improvement on y_{PSE}, the results do shown that Bayesian optimisation can be applied to controllers with a more complex structure than a matrix of PI controllers. It is also possible

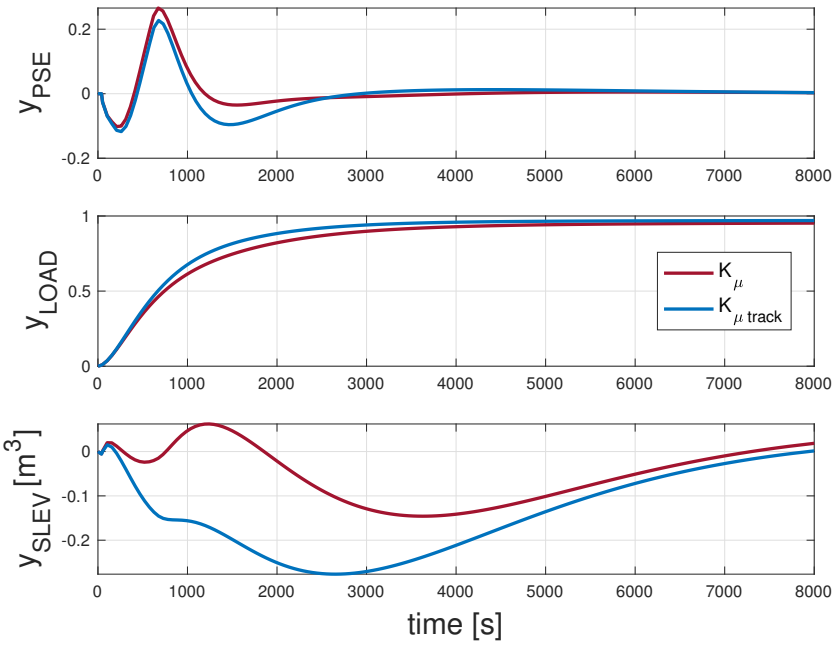

Figure 5.19. Comparison of the set point tracking performance of controllers $\boldsymbol{K}_{\mu}$ and $\boldsymbol{K}_{\mu track}$ in response to a y_{LOAD} set point step change.

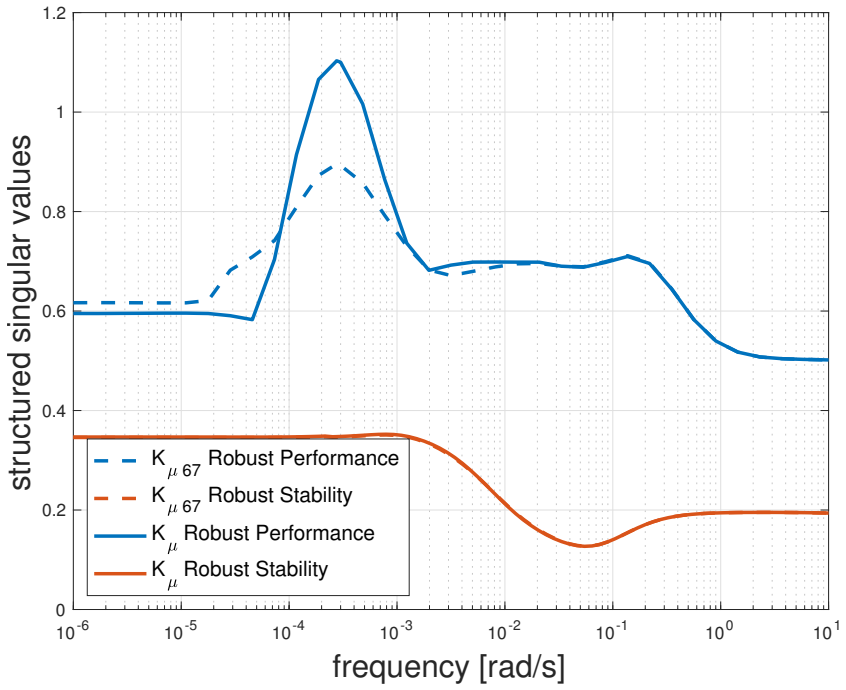

Figure 5.20. Comparison of controllers $\boldsymbol{K}_{\mu}$ and $\boldsymbol{K}_{\mu track}$ μ-plots. The μ-plots of $\boldsymbol{K}_{\mu}$ are represented as dashed lines, while the μ-plots of $\boldsymbol{K}_{\mu track}$ are represented as solid lines.

that there is little opportunity to improve the performance of a μ-synthesised controller before the controller becomes unstable.

Table 5.7. Comparison of the set point tracking properties of controllers $\boldsymbol{K}_{\mu}$ and $\boldsymbol{K}_{\mu track}$. The improvement that $\boldsymbol{K}_{\mu track}$ offers is indicated as a percentage.

Performance criteria	Unit	$\boldsymbol{K}_{\mu}$	$\boldsymbol{K}_{\mu track}$	Improvement (%)
y_{PSE} settling time	seconds	993.59	1017.8	2.38
y_{LOAD} settling time	seconds	27234	61478	55.7
$y_{PSE\ ITAE}$		28815	29717	3.03
$y_{LOAD\ ITAE}$		32645	48760	33.05
$y_{PSE\ RMSE}$		0.22147	0.22215	0.31
$y_{LOAD\ RMSE}$		0.30088	0.31871	5.59

Table 5.8 shows the results of iterations 6 through to 15 of the Bayesian optimisation simulation using objective function (5.22). During simulation, the step test response was evaluated over a period of 2.2 hours to calculate the ITAE. With each Bayesian optimisation iteration only requiring two steps, the 15 iterations as suggested in Table 5.8 would require no less than 66 hours to complete in practice. The best result is found by iteration 10.

5.6 CONCLUSION

The Bayesian optimisation process can be applied to both diagonal and μ-synthesised controllers of an ore milling circuit. While the optimisation of the μ-synthesised controller showed little improvement, there was significant performance improvement over the diagonal controller.

Objective functions can be designed to improve the set point tracking and disturbance rejection performance of an ore milling circuit controller.

The ITAE performance index is found to be suitable for both set point tracking and disturbance rejection. Using the ITAE criteria as performance index has the benefit that the objective function value can be calculated without having to wait for the response of each iteration to settle. Using settling time as a performance index for set point tracking and transient time as a performance index for disturbance rejection is found to be impractical due the long evaluation periods of up to 24 hours waiting for

Table 5.8. Results of Bayesian optimisation simulation using objective function $Q_{\mu track}$, iterations 6 through 15.

Iteration	$Q_{\mu track}$
6	1.9382
7	1.7269
8	1.783
9	1.9244
10	**1.6391**
11	1.6549
12	1.6612
13	1.7342
14	1.784
15	1.6658

transient dynamics to die out. For processes with large time constants, such as the milling circuit, the ITAE based objective function can be of significant benefit, reducing sub-optimal process performance while optimisation is in progress. The inconvenience of using the ITAE is that scaling is required to normalise the contribution of each controlled value, so that each controlled variable contributes equally to the calculated value of the objective function. Calculating the scaling factors requires the overhead of an additional step test.

The constraints of the search domain are calculated by conducting a robust stability analysis. The outcome of the analysis is a range from which samples can be selected that will not result in unstable closed-loop control. The results were confirmed by plotting the robust stability structured singular value over the frequency range of interest and observing that $\mu < 1$ for all frequencies. This approach maximises the constraints of the search domain without introducing parameters that would destabilise closed-loop control. The larger the search domain the better the probability of including the optimal tuning parameters.

Using RMSE as a statistical method to compare the performance of the optimised controllers and reference controllers, Bayesian optimisation is shown to improve both set point tracking as well as

disturbance rejection performance of the diagonal controllers. The improvement of the μ-synthesised controller is not as noteworthy as that of the diagonal controllers possibly as a result of the μ-synthesised controller already being optimised to meet the performance weights as specified in Craig and MacLeod (1996).

The total optimisation period is 60 hours for the diagonal controller and 66 hours for the μ-synthesised controller. Even though Bayesian optimisation has been shown to be capable of improving performance, one needs to consider the feasibility of the long evaluation periods, especially if the evaluation periods result in sub-optimal process performance. Ideally the optimisation process must be automated to step the set points around a point of equilibrium (i.e. positive step change followed by a negative step change) in which case the procedure can be conducted without the supervision of an operator to reset the process after each test. The steps will have to be small enough to remain within the linear region of the process and not disrupt the downstream process but also large enough to rise above the measurement noise.

CHAPTER 6 CONCLUSION

Despite the abundance of researched and published controller tuning methods, the majority of industrial process controllers are poorly tuned. This state of circumstances could be due to the lack of expertise, expense of system identification experiments, expense of domain experts, changing process conditions, and ageing equipment. It is evident that automatic process controller tuners can have a substantial benefit to industry by improving the set point tracking or disturbance rejection performance of controllers.

Considering the need for auto-tuning, this research demonstrates that Bayesian optimisation is a data efficient, model free, on-line tuning method that can optimally tune controllers for industrial processes such as the BTT surge tank and ore milling circuit. The Gaussian process surrogate model, based on the Matérn parameter 5/2 covariance function, minimised using the expected improvement acquisition function is shown to be suitable choices for the Bayesian optimisation of industrial process controllers.

Objective functions can be designed to promote either set point tracking or disturbance rejection of controllers. Objective functions can be based on multiple performance criteria, scaled to contribute equally to the objective function value, or weighted to promote the performance of a favoured process variable over another.

The constraints of the search domain can be determined analytically, by conducting a robust stability analysis on the closed-loop system consisting of a controller with uncertain tuning parameters. This methods expands the search domain to the threshold of instability thereby improving the probability of including the optimal parameters while excluding unstable parameters. The use of robust stability

analysis requires a process model on which to conduct the analysis. If such a model is not available from literature, a linear model can be approximated by conducting system identification experiments.

Using the RMSE to statistically evaluate the improvement that Bayesian optimisation offers, results demonstrate that decentralised PI and inverse multivariable controllers can be optimised on the industrial processes presented. The improvement of the μ-controller is marginal, but the results do demonstrate that Bayesian optimisation can be used to search for optimal pole positions.

Based on the auto-tuning results, Bayesian optimisation is well suited for the auto-tuning of industrial PI controllers in a decentralised or inverse multivariable controller structure. The optimisation of processes with smaller time constants than the BTT surge tank and ore milling circuit is expected to perform event better, since the smaller time constants will result in shorter iteration periods and faster conversion rates. BO can be used to auto-tune controllers during commissioning or during operation when poor controller performance is observed as a result of changing process conditions and ageing equipment.

Opportunities for future work in the field of Bayesian optimisation of process controllers include:

- Minimising the impact that auto-tuning has on sub-optimal production performance and the resulting loss of revenue.
- Applying Bayesian optimisation to model predictive controllers.
- Comparing the performance of Bayesian optimisation to the multivariable relay method presented by Wang, Zou, Lee and Bi (1997).

Provided that future work can minimise sub-optimal process performance during optimisation, Bayesian optimisation shows potential to automatically tune industrial process controllers.

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. (2004). Revisiting the Ziegler-Nichols step response method for PID control, *Journal of Process Control* **14**(6): 635–650.

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.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization., *Journal of Machine Learning Research* **13**(2): 281–305.

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

Berner, J., Soltesz, K., Hägglund, T. and Åström, K. J. (2018). An experimental comparison of PID autotuners, *Control Engineering Practice* **73**: 124–133.

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.

Bristol, E. (1966). On a new measure of interaction for multivariable process control, *IEEE Transactions on Automatic Control* **11**(1): 133–134.

Brochu, E., Cora, V. and Freitas, N. (2010). 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, Technical Report UBC* .

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

Burchell, J., le Roux, J. and Craig, I. (2023). Nonlinear model predictive control for improved water recovery and throughput stability for tailings reprocessing, *Control Engineering Practice* **131**: 105385.

Coetzee, L. C., Craig, I. K. and Kerrigan, E. C. (2010). Robust nonlinear model predictive control of a run-of-mine ore milling circuit, *IEEE Transactions on Control Systems Technology* **18**(1): 222–229.

Cohen, G. H. and Coon, G. A. (1953). Theoretical consideration of retarded control, *Transactions of the ASME* **75**: 827–834.

Craig, I., Hulbert, D., Metzner, G. and Moult, S. (1992a). Extended particle-size control of an industrial run-of-mine milling circuit, *Powder Technology* **73**(3): 203–210.

Craig, I. K., Hulbert, D. G., Metzner, G. and Moult, S. P. (1992b). Optimised multivariable control of an industrial run-of-mine milling circuit, *Journal of the Southern African Institute of Mining and Metallurgy* **92**(6): 169–176.

Craig, I. K. and MacLeod, I. M. (1995). Specification framework for robust control of a run-of-mine ore milling circuit, *Control Engineering Practice* **3**(5): 621–630.

Craig, I. K. and MacLeod, I. M. (1996). Robust controller design and implementation for a run-of-mine ore milling circuit, *Control Engineering Practice* **4**(1): 1–12.

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

Dogru, O., Velswamy, K., Ibrahim, F., Wu, Y., Sundaramoorthy, A. S., Huang, B., Xu, S., Nixon, M. and Bell, N. (2022). Reinforcement learning approach to autonomous PID tuning, *Computers & Chemical Engineering* **161**: 107760.

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, *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 5850–5856.

Galán, O., Barton, G. and Romagnoli, J. (2002). Robust control of a SAG mill, *Powder Technology* **124**(3): 264–271.

Garcia, C. E. and Morari, M. (1982). Internal model control. A unifying review and some new results, *Industrial & Engineering Chemistry Process Design and Development* **21**(2): 308–323.

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

Hang, C. C., Loh, A. P. and Vasnani, V. U. (1994). Relay feedback auto-tuning of cascade controllers, *IEEE Transactions on Control Systems Technology* **2**(1): 42–45.

Hang, C.-C., Wang, Q. and Cao, L.-S. (1995). Self-tuning Smith predictors for processes with long dead time, *International Journal of Adaptive Control and Signal Processing* **9**(3): 255–270.

He, M., Cai, W., Wu, B. and He, M. (2005). Simple decentralized PID controller design method based on dynamic relative interaction analysis, *Industrial & Engineering Chemistry Research* **44**: 8334–8344.

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.

Huang, H.-P., Jeng, J.-C. and Luo, K.-Y. (2005). Auto-tune system using single-run relay feedback test and model-based controller design, *Journal of Process Control* **15**(6): 713–727.

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.

Karageorgos, J., Genovese, P. and Baas, D. (2006). Current trends in SAG and AG mill operability and control, *Proceedings of an International Conference on Autogenous and Semiautogenous Grinding Technology*, Vol. 3, pp. 191–206.

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.

Lam, R., Poloczek, M., Frazier, P. and Willcox, K. E. (2018). Advances in Bayesian optimization with applications in aerospace engineering, *2018 AIAA Non-Deterministic Approaches Conference*, p. 1656.

Lawrence, N. P., Stewart, G. E., Loewen, P. D., Forbes, M. G., Backstrom, J. U. and Gopaluni, R. B. (2020). Optimal PID and antiwindup control design as a reinforcement learning problem, *IFAC-PapersOnLine* **53**(2): 236–241.

Le Roux, J. D. and Craig, I. K. (2019). Plant-wide control framework for a grinding mill circuit, *Industrial & Engineering Chemistry Research* **58**(26): 11585–11600.

Le Roux, J. D., Craig, I. K., Hulbert, D. and Hinde, A. (2013). Analysis and validation of a run-of-mine ore grinding mill circuit model for process control, *Minerals Engineering* **43**: 121–134.

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.

Luyben, W. L. (1990). *Process Modeling, Simulation and Control for Chemical Engineers*, 2nd edn, McGraw-Hill Publishing Company.

MATLAB (2022). *version 9.12.0 (R2022a)*, The MathWorks Inc., Natick, Massachusetts.

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

Moriconi, R., Deisenroth, M. P. and Sesh Kumar, K. (2020). High-dimensional Bayesian optimization using low-dimensional feature spaces, *Machine Learning* **109**(9): 1925–1943.

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.

Pongfai, J., Su, X., Zhang, H. and Assawinchaichote, W. (2020). PID controller autotuning design by a deterministic Q-SLP algorithm, *IEEE Access* **8**: 50010–50021.

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., Burchell, J., Olivier, L. and Craig, I. (2021). Towards an access economy model for industrial process control: A bulk tailings treatment plant case study, *IFAC-PapersOnLine* **54**(21): 121–126.

Rokebrand, L. L., Burchell, J. J., Olivier, L. E. and Craig, I. K. (2020). Competing advanced process control via an industrial automation cloud platform, *arXiv preprint arXiv:2011.13184.* .

Seborg, D. E., Edgar, T. F., Mellichamp, D. A. and Doyle, F. J. (2011). *Process dynamics and control*, John Wiley & Sons.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization, *Proceedings of the IEEE* **104**(1): 148–175.

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.

Skogestad, S. and Postlethwaite, I. (2007). *Multivariable feedback control: analysis and design*, Vol. 2, Wiley.

Snoek, J., Larochelle, H. and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms, *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 2*, pp. 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* pp. 997–1005.

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z. and Guyon, I. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020, *NeurIPS 2020 Competition and Demonstration Track*, PMLR, pp. 3–26.

Wang, Q.-G., Hang, C.-C. and Bi, Q. (1997). Process frequency response estimation from relay feedback, *Control Engineering Practice* **5**(9): 1293–1302.

Wang, Q.-G., Hang, C.-C. and Zou, B. (1997). Low-order modeling from relay feedback, *Industrial & Engineering Chemistry Research* **36**(2): 375–381.

Wang, Q.-G., Zou, B., Lee, T.-H. and Bi, Q. (1997). Auto-tuning of multivariable PID controllers from decentralized relay feedback, *Automatica* **33**(3): 319–330.

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.

Ziegler, J. G. and Nichols, N. B. (1942). Optimum setting for automatic controllers, *Transactions of the ASME* **64**: 759–768.