

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 \mathbf{K}_{track} . \mathbf{K}_{track} is the best result of optimising \mathbf{K}_α 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

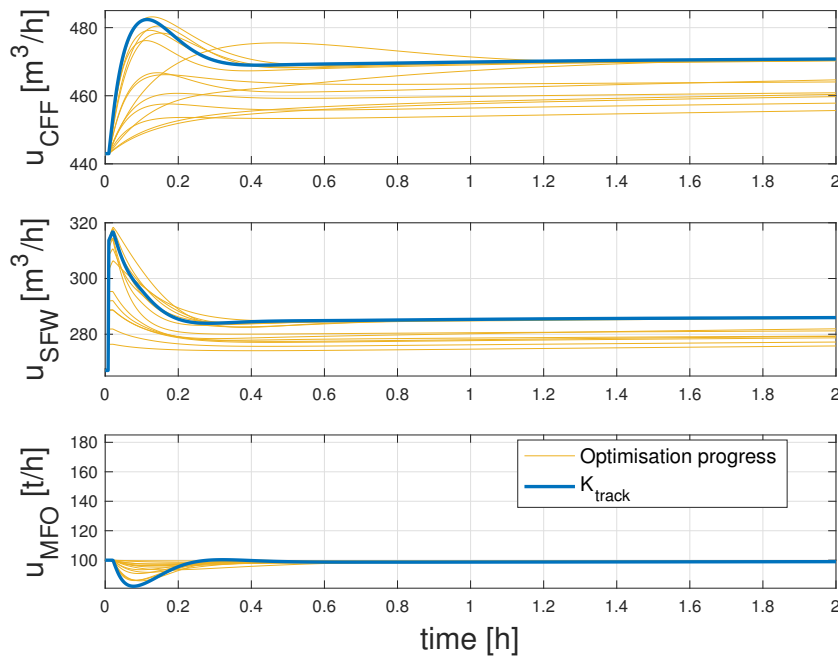


Figure 5.4. Response of the manipulated variables to a y_{PSE} set point step change during Bayesian optimisation using objective function Q_{Track} .

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 K_{α} and the controller retuned using Bayesian optimisation 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 K_{track} and K_{α} . The improvement that controller K_{track} offers is indicated as a percentage.

Performance	K_{track}	K_{α}	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

$$k_{P11} = -50.772, \tau_{I11} = 2.7411 \quad (5.13a)$$

$$k_{P22} = 466.81, \tau_{I22} = 0.15388 \quad (5.13b)$$

$$k_{P33} = 1144.2, \tau_{I33} = 21.105. \quad (5.13c)$$

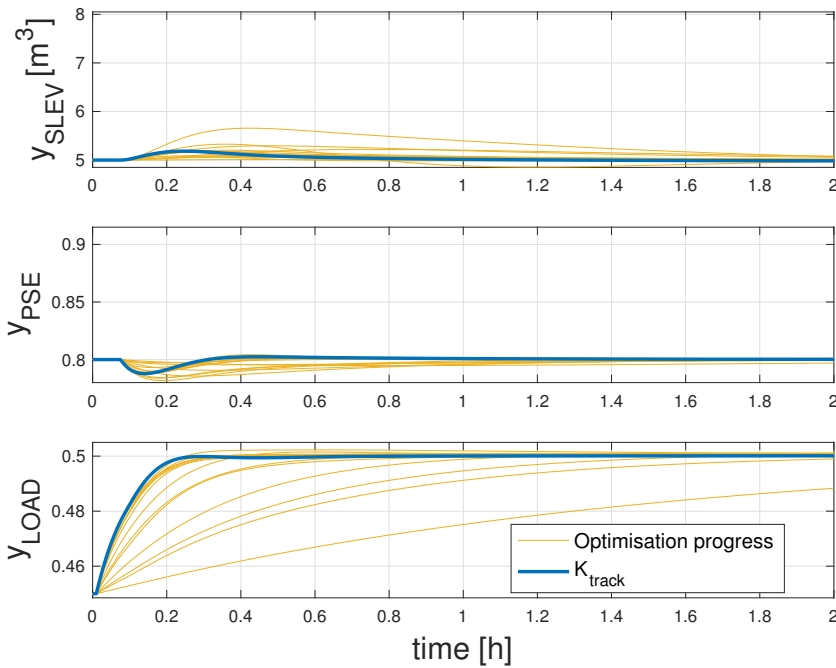


Figure 5.5. Response of the controlled variables to a y_{LOAD} set point step change during Bayesian optimisation using objective function Q_{track} .

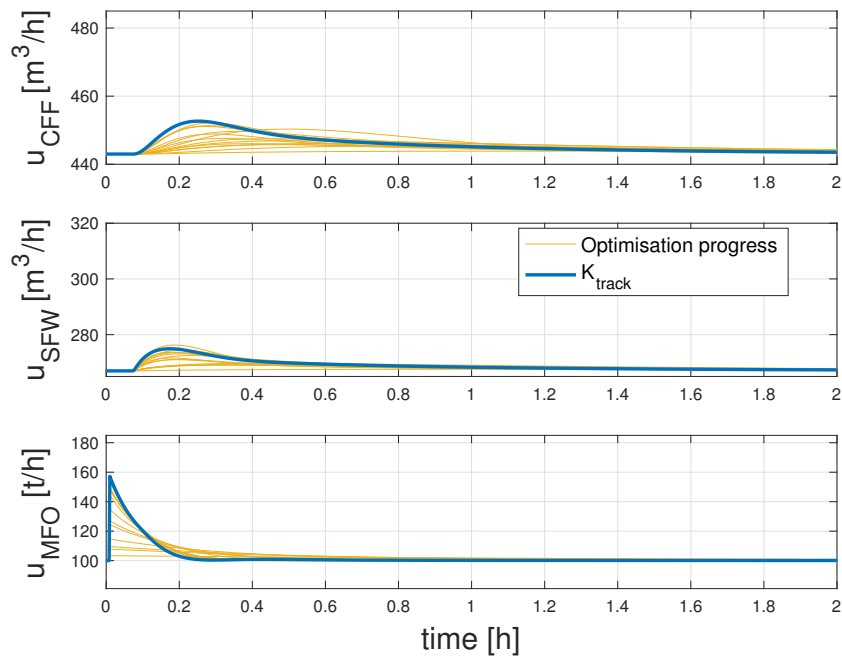


Figure 5.6. Response of the manipulated variables to a y_{LOAD} set point step change during Bayesian optimisation using objective function Q_{track} .

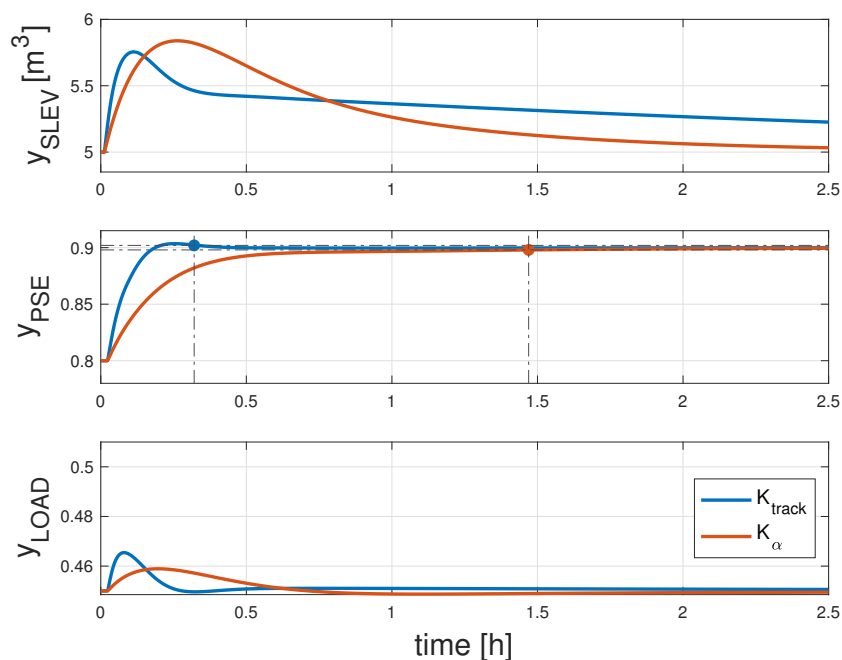


Figure 5.7. Comparison of the set point tracking performance of controllers K_{track} and K_{α} in response to a y_{PSE} set point step change. The markers indicate the settling time of the responses.

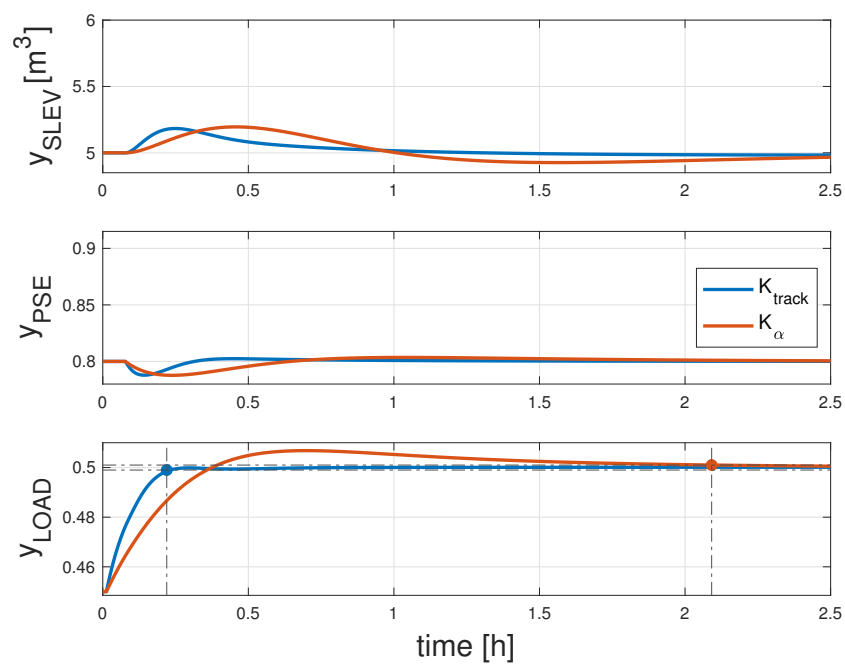


Figure 5.8. Comparison of the set point tracking performance of controllers K_{track} and K_{α} in response to a y_{LOAD} set point step change. The markers indicate the settling time of the responses.

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. \mathbf{K}_{reject} is the best result of optimising \mathbf{K}_α 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 iteration 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.

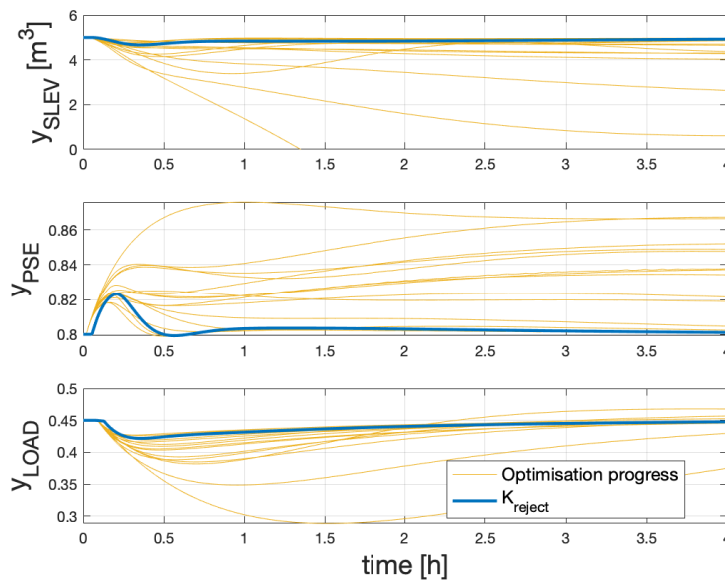


Figure 5.9. Response of the controlled variables to an ore hardness step change during Bayesian optimisation using objective function Q_{reject} .

Fig. 5.11 compares the disturbance response of \mathbf{K}_{reject} and \mathbf{K}_α 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}

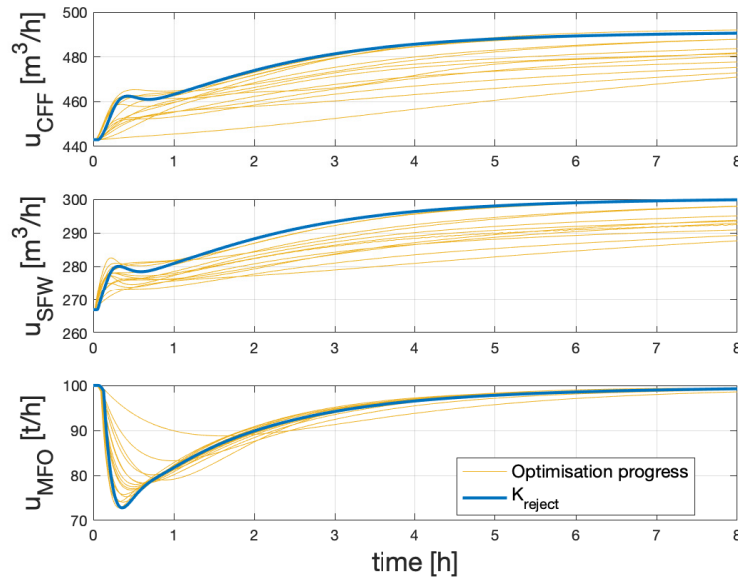


Figure 5.10. Response of the manipulated variables to an ore hardness step change during Bayesian optimisation using objective function Q_{reject} .

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

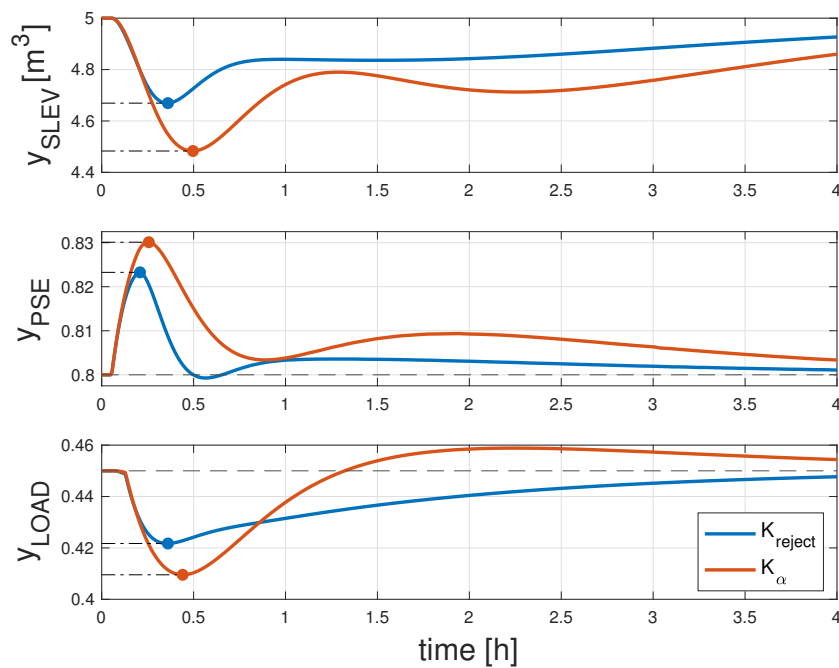


Figure 5.11. Comparison of the disturbance rejection performance of controllers K_{reject} and K_{α} in response to an ore hardness step change. The markers indicate the peak disturbance error of the responses.

Table 5.5. Comparison of disturbance rejection properties of controllers K_{reject} and K_{α} . The improvement that controller K_{reject} offers is indicated as a percentage.

Performance	K_{reject}	K_{α}	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

$$k_{P11} = -45.607, \tau_{I11} = 0.68394 \quad (5.14a)$$

$$k_{P22} = 247.97, \tau_{I22} = 0.11517 \quad (5.14b)$$

$$k_{P33} = 956.8, \tau_{I33} = 32.947. \quad (5.14c)$$

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 $\mathbf{y} = \mathbf{G}(s)\mathbf{u}$, determined from multiple step tests conducted by Craig and MacLeod (1996) is

$$\begin{bmatrix} y_{PSE} \\ y_{LOAD} \\ y_{SLEV} \end{bmatrix} = \begin{bmatrix} \frac{0.14}{175s+1}e^{-40s} & \frac{-0.082}{1766s+1}e^{-620s} & \frac{-0.0575}{167s+1}e^{-40s} \\ 0 & \frac{2.21 \times 10^{-5}}{s} & 0 \\ \frac{0.00253}{s} & 0 & \frac{-0.00299}{s} \end{bmatrix} \begin{bmatrix} u_{SFW} \\ u_{MFS} \\ u_{CFE} \end{bmatrix}. \quad (5.15)$$

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 $\mathbf{K}_{\mu 67}$, 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

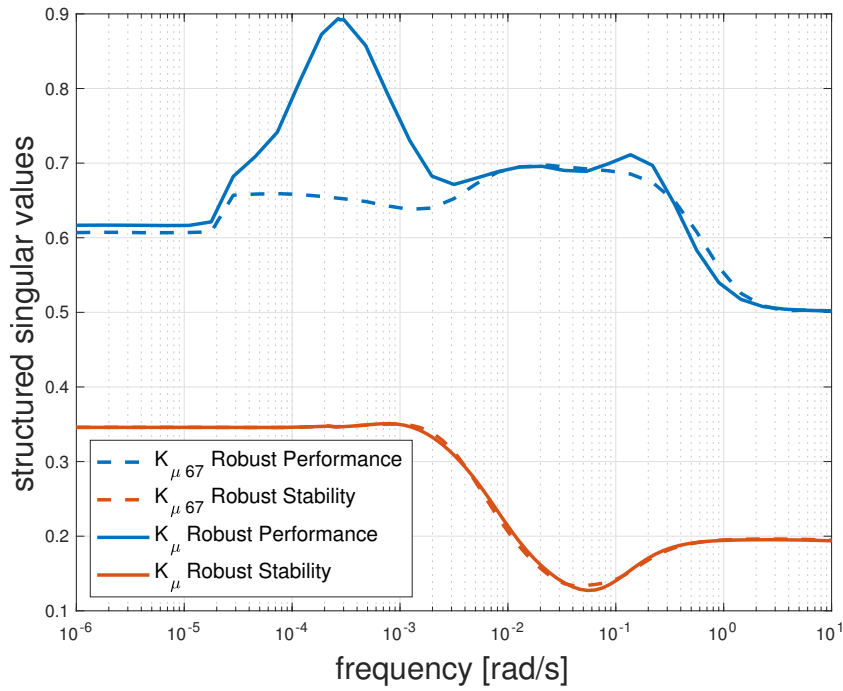


Figure 5.12. Robust stability and robust performance structured singular value plots of the $\mathbf{K}_{\mu 67}$ and \mathbf{K}_{μ} controller. The plots of $\mathbf{K}_{\mu 67}$ are trended as dashed lines, while the plots of \mathbf{K}_{μ} are solid lines.

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

$$\mathbf{K}_{\mu} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \quad (5.16)$$

where each matrix entry has the form

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

Fig. 5.12 compares the structured singular values μ for robust performance and robust stability of $\mathbf{K}_{\mu 67}$ and reduced order controller \mathbf{K}_{μ} 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. \mathbf{K}_{μ} will therefore remain robustly stable but will not be able to achieve the same level of performance as $\mathbf{K}_{\mu 67}$ for the modelled uncertainty.

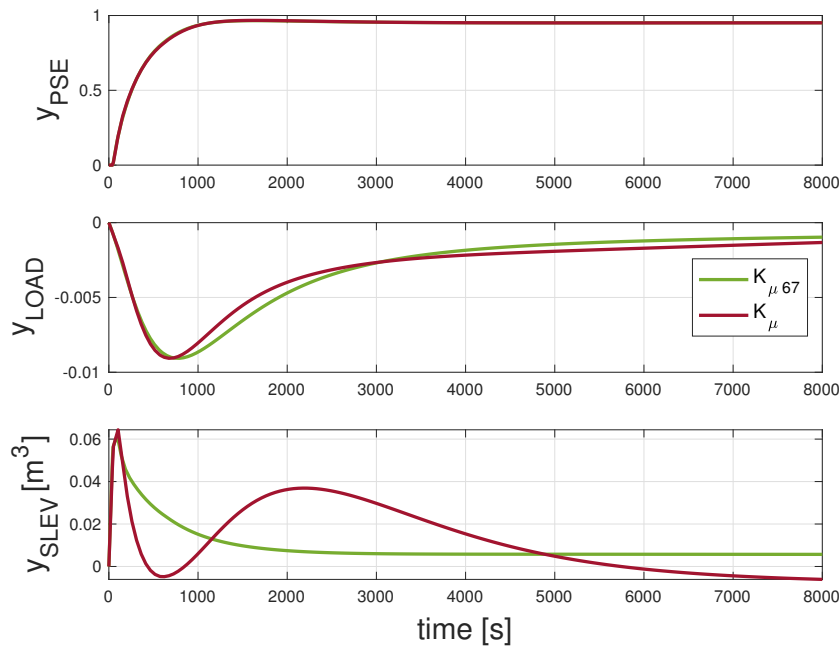


Figure 5.13. Comparison of the set point tracking performance of controllers $K_{\mu 67}$ and K_{μ} in response to a y_{PSE} set point step change.

Figs. 5.13 and 5.14 compare the responses of controller $K_{\mu 67}$ 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 K_{μ} compare well to that of $K_{\mu 67}$, but K_{μ} is not able to suppress the interaction with y_{SLEV} as successfully as $K_{\mu 67}$.

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 K_{μ} compare well to that of $K_{\mu 67}$. Once again controller K_{μ} does not manage to suppress the interaction with y_{SLEV} as well as $K_{\mu 67}$.

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.

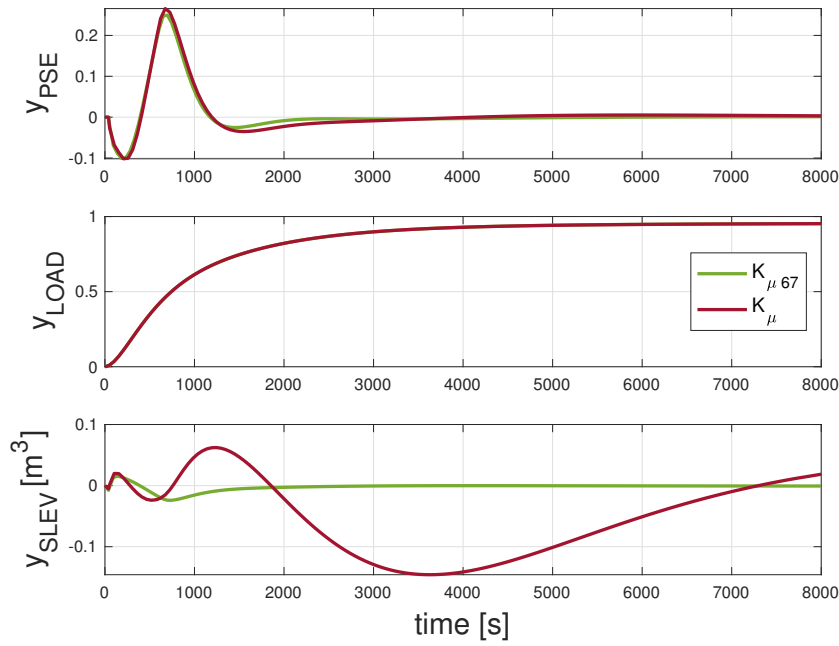


Figure 5.14. Comparison of the set point tracking performance of controllers $\mathbf{K}_{\mu 67}$ and \mathbf{K}_{μ} in response to a y_{LOAD} set point step change.

5.5.3 Optimisation of set point tracking

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

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

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

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

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

The order of \mathbf{K}_{μ} has been reduced to 10, therefore \mathbf{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), \mathbf{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 \mathbf{A} have negative real parts, that is, lie in the open left-half plane (Skogestad and Postlethwaite, 2007). The eigenvalues of \mathbf{A} are equivalent to the poles of the transfer function and therefore adjusting the eigenvalues of \mathbf{A} is equivalent to changing the positions of the controller poles. The controller \mathbf{K}_μ 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 \mathbf{A} , and poles of controller \mathbf{K}_μ , are

$$\mathbf{A} = \text{diag}(a_1, a_2, \dots, a_{10}). \quad (5.19)$$

where

$$a_1 = -0.286 \quad (5.20a)$$

$$a_2 = -0.0474 \quad (5.20b)$$

$$a_3 = -0.0394 \quad (5.20c)$$

$$a_4 = -0.00543 \quad (5.20d)$$

$$a_5 = -0.00382 \quad (5.20e)$$

$$a_6 = -0.00071 \quad (5.20f)$$

$$a_7 = -0.00025 \quad (5.20g)$$

$$a_8 = -0.000115 \quad (5.20h)$$

$$a_9 = -5.31 \times 10^{-05} \quad (5.20i)$$

$$a_{10} = -2.75 \times 10^{-05}. \quad (5.20j)$$

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 \mathbf{K}_μ , 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 \mathbf{K}_μ 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

$$a_1 \in [-0.301, -0.272] \quad (5.21a)$$

$$a_2 \in [-0.0497, -0.045] \quad (5.21b)$$

$$a_3 \in [-0.0413, -0.0374] \quad (5.21c)$$

$$a_4 \in [-0.0057, -0.00516] \quad (5.21d)$$

$$a_5 \in [-0.00401, -0.00363] \quad (5.21e)$$

$$a_6 \in [-0.000746, -0.000675] \quad (5.21f)$$

$$a_7 \in [-0.000263, -0.000238] \quad (5.21g)$$

$$a_8 \in [-0.000121, -0.000109] \quad (5.21h)$$

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

$$a_{10} \in [-2.88 \times 10^{-05}, -2.61 \times 10^{-05}]. \quad (5.21j)$$

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 \mathbf{G} and \mathbf{K}_μ 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 \mathbf{K}_μ . 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 \mathbf{K}_μ must be optimally placed such that the settling time is reduced after a set point change.

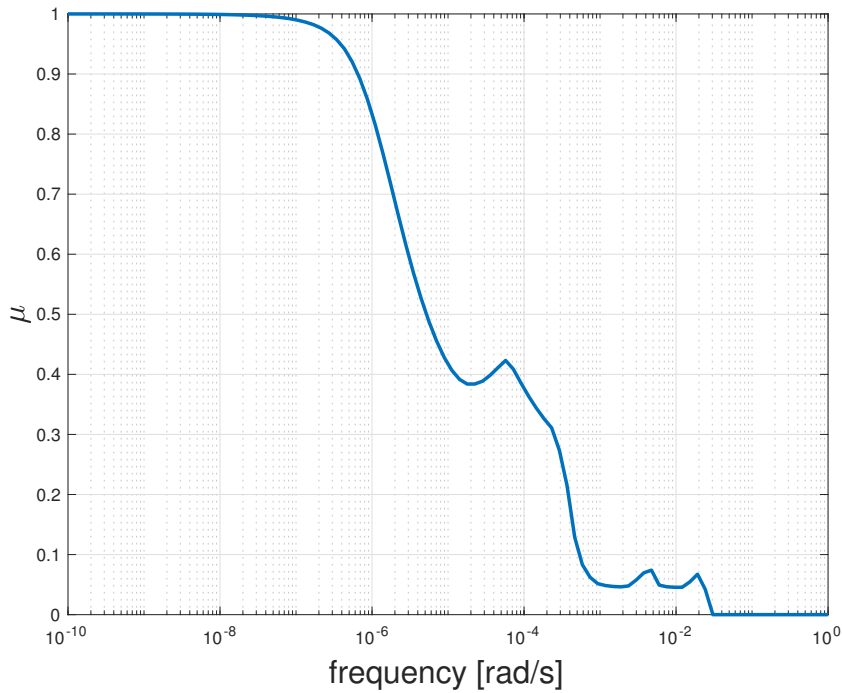


Figure 5.15. Robust stability structured singular value (μ) plot with poles constrained as per (5.21).

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

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

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 $\omega_1 = \omega_2 = 1$.

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 \mathbf{K}_μ 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 $\mathbf{K}_{\mu track}$ is highlighted. Fig. 5.17 shows that during one of the iterations, y_{SLEV} deviates by more than 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 \mathbf{K}_μ and $\mathbf{K}_{\mu track}$. Controller \mathbf{K}_μ is the reduced order controller and $\mathbf{K}_{\mu track}$ 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

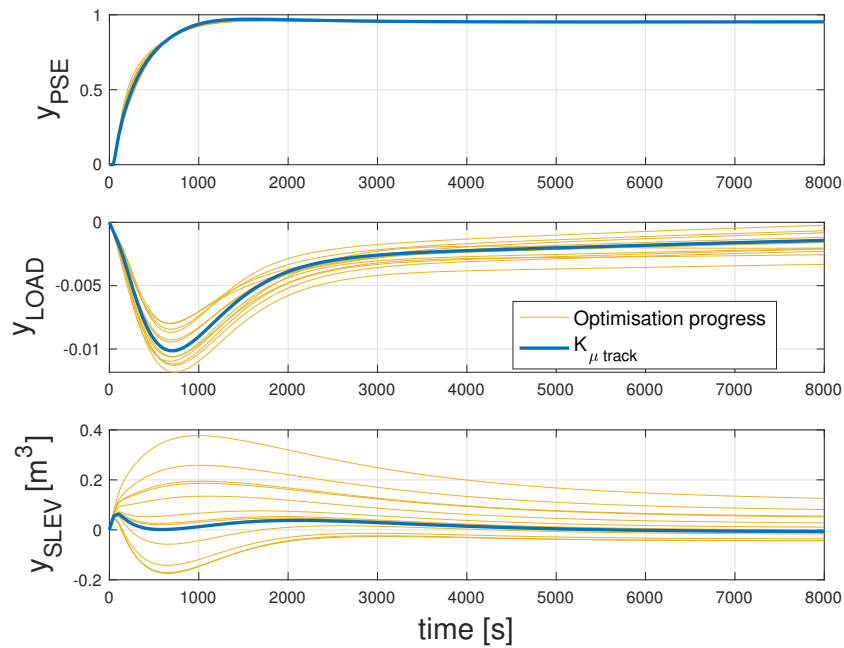


Figure 5.16. Response of the controlled variables to a y_{PSE} set point step change during Bayesian optimisation using objective function $Q_{\mu track}$.

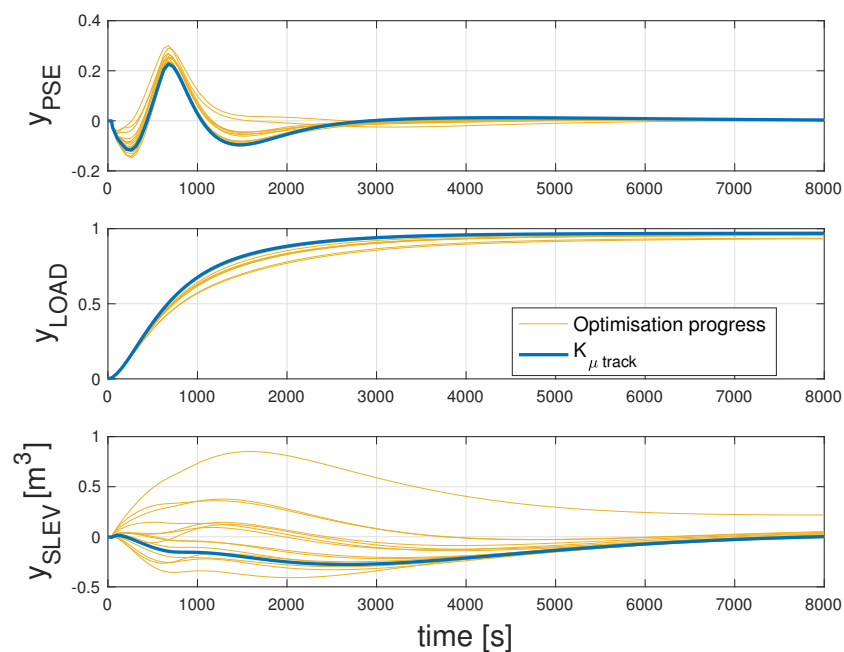


Figure 5.17. Response of the controlled variables to a y_{LOAD} set point step change during Bayesian optimisation using objective function $Q_{\mu track}$.

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} .

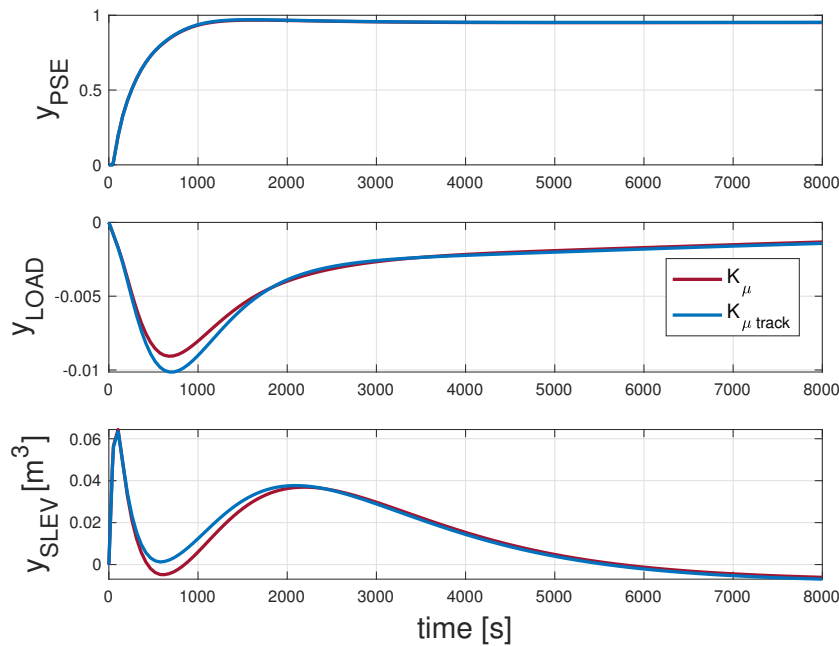


Figure 5.18. Comparison of the set point tracking performance of controllers \mathbf{K}_μ and $\mathbf{K}_{\mu track}$ in response to a y_{PSE} set point step change.

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 \mathbf{K}_μ to that of $\mathbf{K}_{\mu track}$. The robust stability of the two controllers is similar but the robust performance of $\mathbf{K}_{\mu track}$ 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 \mathbf{K}_μ and $\mathbf{K}_{\mu track}$. 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 show that Bayesian optimisation can be applied to controllers with a more complex structure than a matrix of PI controllers. It is also possible

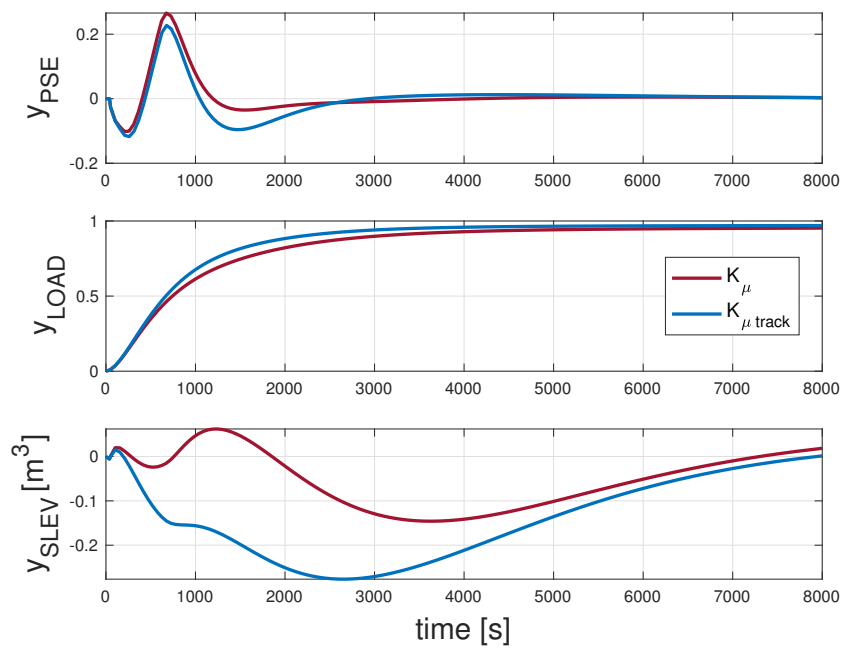


Figure 5.19. Comparison of the set point tracking performance of controllers K_μ and $K_{\mu track}$ in response to a y_{LOAD} set point step change.

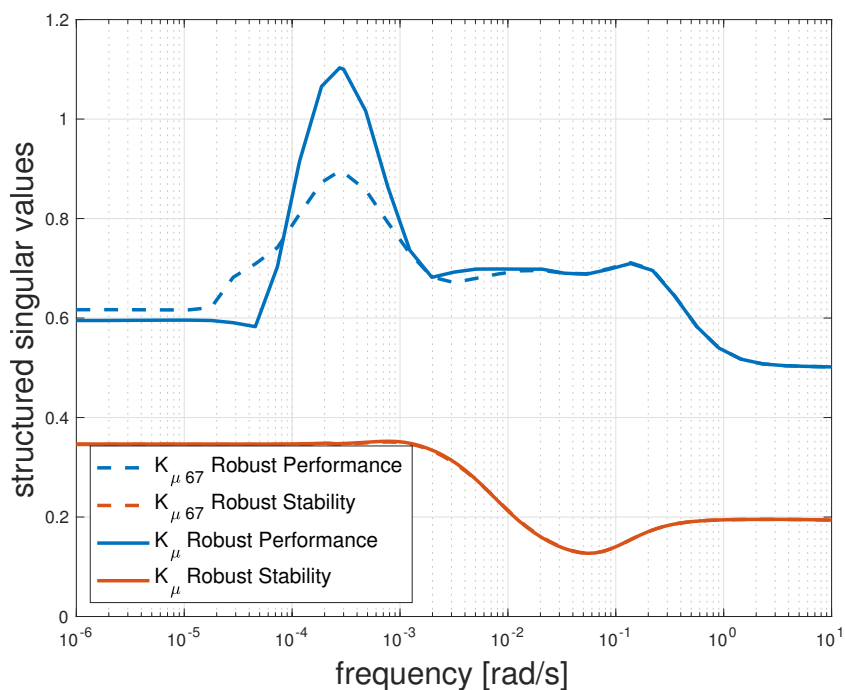


Figure 5.20. Comparison of controllers K_μ and $K_{\mu track}$ μ -plots. The μ -plots of K_μ are represented as dashed lines, while the μ -plots of $K_{\mu track}$ are represented as solid lines.

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 \mathbf{K}_μ and $\mathbf{K}_{\mu track}$. The improvement that $\mathbf{K}_{\mu track}$ offers is indicated as a percentage.

Performance criteria	Unit	\mathbf{K}_μ	$\mathbf{K}_{\mu track}$	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 method 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 even 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.

REFERENCES

- 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 Moul, 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 Moul, 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.

REFERENCES

- 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.

REFERENCES

- 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.

REFERENCES

- 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.

REFERENCES

- 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.