# Enhancing PI Tuning for Plant Commissioning Using Transfer Learning and Bayesian Optimization

Boulaid Boulkroune, Joachim Verhelst, Branimir Mrak, Bruno Depraetere and Joram Meskens and Pieter Bovijn

*Flanders Make vzw, Oude Diestersebaan 133, 3920 Lommel, Belgium*

*ﬂ*

Keywords: Controller Tuning, Transfer Learning, Bayesian Optimization, Thermal Plant.

Abstract: A novel approach for accelerating the auto-tuning of PI controllers during the commissioning phase is proposed in this study. This approach combines transfer learning and Bayesian optimization (BO) to minimize the number of iterations required to converge to the optimal solution. Transfer learning is employed to extract valuable information from available historical data derived from expert tuning of other equivalent process variants. In the absence of historical data, a simulation model can also be utilized to generate data from different model variants (e.g., changing the value of unknown parameters). In this study, a simulation model is used for generating historical data. The approach's efficiency is demonstrated through its application to a thermal plant, achieving a significant reduction in the number of iterations required to reach the optimizer's optimal solution.



## 1 INTRODUCTION

Proportional-Integral-Derivative (PID) controllers are the most commonly used controllers in industrial processes, despite the availability of numerous advanced controller variants in academia. The popularity of these controllers can be attributed to their simplicity, robustness, ease of use, and the fact that they do not require model knowledge. Numerous manual tuning methods involve monitoring the process response following adjustments to the controller setpoint are proposed. However, PID tuning remains a tedious and time-consuming process, and it is generally very difficult to achieve optimal performance when done manually.

Several auto-tuning approaches have been proposed in the literature with the aim of automating the process of selecting appropriate controller parameters (for instance, (Aidan, 2006), (Liuping, 2017), (Ho et al., 2003)). Broadly speaking, these methodologies can be classified into data-driven and model-based approaches. Data-driven controller design (see, for instance, (Bazanella et al., 2011)) offers the advantage of not requiring prior model knowledge. Notable works in this area include Virtual Reference Feedback Tuning (VRFT) (Campi et al., 2002), Fictitious Reference Iterative Tuning (FRIT) (Hjalmarsson, 1998), Iterative Feedback Tuning (IFT) (Kaneko et al., 2005), a one-step tuning scheme for a 2DOF control system (Kinoshita and Yamamoto, 2018), and step response-based methods (Sanchis and Peñarrocha-Alós, 2022). While data-driven PID tuning approaches offer numerous advantages, they are not without their drawbacks. These drawbacks include data dependency, limited generalization to diverse operating conditions, sensitivity to noisy data, and the risk of overfitting when the process diverges from the data distribution.

The second category consists of model-based approaches, which include a variety of techniques. For more in-depth information, readers are referred to sources such as (Boyd et al., 2016) and related references. Model-based methods demonstrate their full potential when accurate models are available. However, in practice, developing precise and robust models is often difficult due to the lack of practical and comprehensive descriptions of uncertainties (Gevers, 2002).

An alternative solution is to use a hybrid approach that combines data-driven and model-based methods, as proposed in (Fujimoto et al., 2023). This approach is formulated within the Bayesian optimization framework, where the process model is used to define the prior mean function. While promising, the effectiveness of this approach heavily depends on the accuracy of the underlying model. Inaccurate models may lead to convergence toward local minima, par-



Boulkroune, B., Verhelst, J., Mrak, B., Depraetere, B., Meskens, J. and Bovijn, P.
Enhancing PI Tuning for Plant Commissioning Using Transfer Learning and Bayesian Optimization.
DOI: 10.5220/0013712600003982

In *Proceedings of the 22nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2025) - Volume 1*, pages 235-242
ISBN: 978-989-758-770-2; ISSN: 2184-2809

ticularly if the optimization process emphasizes exploitation over exploration. A similar approach is proposed by (Boulkroune et al., 2024), where the method utilizes both process model knowledge and Bayesian optimization capabilities. Initially, the model is refined by identifying unknown or uncertain parameters. The improved model is then used in simulations to search for an optimal configuration of the PI controller. The results obtained—including the initial estimate, upper and lower bounds, and the Gaussian process mean—are subsequently used to initialize the Bayesian optimization process during the commissioning phase. By properly initializing the Bayesian optimization, the number of iterations required to reach the optimizer's optimal solution can be significantly reduced.

In (Reynoso-Meza et al., 2014), a hybrid multi-objective optimization method for tuning PI controllers is introduced, with a particular focus on reliability-based optimization scenarios. The study employs Monte Carlo simulations to quantitatively evaluate performance degradation of the controller due to unforeseen or unmodeled system dynamics. The utilization of a multi-objective framework adds a level of complexity for users, requiring proficiency in multi-objective optimization techniques.

Other auto-tuning approaches have been proposed based on reinforcement learning (RL), where controller parameters are adjusted through a training process (see, for example, (Lu et al., 2017), (Doerr et al., 2017)). These methods exploit RL's capacity to learn optimal control policies through interaction with the environment, making them well-suited for dynamic and uncertain systems. In contrast, the approach in (Shipman, 2021) focuses on training a combination of value and policy functions, rather than directly tuning controller gains. This strategy aims to enhance controller stability and performance by optimizing the underlying decision-making process. Overall, these advanced auto-tuning techniques present promising solutions for improving the performance and reliability of PI controllers across a range of industrial applications. However, they also introduce added complexity and demand a deeper understanding of optimization and machine learning principles.

In this paper, we present an approach similar to that of (Boulkroune et al., 2024), incorporating several key enhancements. We assume the availability of historical data, which may either originate from other processes or be generated through simulation models. The simulation data can come from different model variants, offering a diverse set of insights. To extract valuable knowledge from this historical data, we employ the two-stage transfer learning approach proposed in (Li et al., 2022). This method is particularly effective in managing the complementary nature of source tasks (historical data) and system dynamics during the knowledge aggregation process, thereby facilitating more efficient information transfer across different tasks. The proposed approach is tested using data from a thermal plant setup. The collected experimental data, which was originally gathered for different purposes, is fitted to a Gaussian Process (GP) model. This model serves as a surrogate for the actual plant, eliminating the need for additional costly and time-consuming experiments. Additionally, a simplified thermal model is used to generate historical data by varying the values of some unknown parameters (e.g., cupper mass) within specified ranges.

This paper is organized as follows: In Section 2, we introduce the preliminary concepts and define the problem. Section 3 presents the main approach, focusing on how it accelerates the PI controller tuning process during the commissioning phase. Section 4 demonstrates the application of this approach to auto-tune a PI controller for a thermal plant. Finally, Section 5 provides the conclusions and summarizes the key takeaways from the study.

## 2 PRELIMINARY AND PROBLEM FORMULATION

The auto-tuning of a PI controller in a real plant is the focus of this paper. A simple scheme of the PI controller is shown in Fig. 2. The signal $y^{ref}$ is the reference or set point to be tracked. $u$ and $y$ are respectively the input and output of the process. $\omega$ represents the process perturbation, and $\nu$ denotes the measurement noise. $e = y^{ref} - y$ is the tracking error.

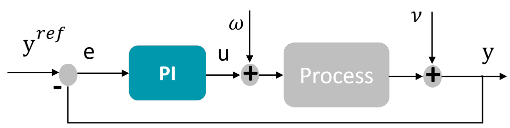



Figure 1: PI controller.

The PI controller has the following structure:

$$u_t = K_p e(t) + K_I \int e(t)\,\mathrm{d}t \tag{1}$$

where $K_p$ and $K_i$ represents the proportional and integral gains. An anti-windup measure is also considered to ensure the actuator constraints.

To tune the controller gains ($K_p$ and $K_i$), we need to define a cost function based on the desired tracking performance of the PI controller (i.e., overshoot, settling time, integral absolute error (IAE), integral time absolute error (ITAE), etc.). In this paper, a linear combination of commonly used control key performance indicators (KPIs): settling time, overshoot, and IAE is defined in the following equations:

$$J_\theta = a_1 Osh(\theta) + a_2 St(\theta) + a_3 IAE(\theta) \qquad (2)$$

where $a_i, i = 1:3$, are known and constant weights. $\theta$ is a vector that contains the control parameters. $Osh$ and $St$ are the overshoot and settling time, respectively. $IAE$ represents the Integrated Absolute Error and is given by: $IAE = \int_0^\infty | e(t) | \, dt$.

This specific set of KPIs is chosen to cover typical dynamic responses of interest, such as settling when changing references (settling time), minimizing error to reference (IAE), and limiting overshoot, as many thermal plants are particularly sensitive to very low overshoot. The weights are selected to provide a quantifiable performance from the step response, based on qualitative evaluation by a plant expert.

The auto-tuning PI controller is now formulated as a black-box multi-objective optimization problem given by:

$$\begin{aligned} \min_{\theta} \quad & J_\theta \\ \text{s.t.} \quad & \theta = [K_p, K_I], \theta^{up} \leq \theta \leq \theta^{lp} \end{aligned} \qquad (3)$$

$\theta^{up}$ and $\theta^{lp}$ are respectively the upper and lower bounds of each element in the vector $\theta$. The primary objective at this juncture is to find the optimal solution for the challenging optimization problem (3). Given the absence of a closed-form expression for the objective function and its expensive evaluation (or sampling), the Bayesian optimization approach emerges as the most appropriate choice. The costly nature of objective function evaluation is attributed to the intended tuning on the actual process during the commissioning phase. In Figure 2, we illustrate the execution of the optimization framework using the BO approach. Due to space limitations, we refrain from providing in-depth technical details about the BO optimization approach. Readers seeking a comprehensive explanation are encouraged to refer to the relevant literature, such as (Garnett, 2023), for a more detailed understanding. Unfortunately, even though the BO approach is a promising technique, its performance faces efficiency challenges due to the limited number of configuration evaluations possible within a constrained budget. Moreover, incorporating available historical data is not straightforward. This problem is very challenging, mainly due to the difficulties in extracting source knowledge from available data and aggregating and transferring this knowledge to the new target domain. In the literature, this problem is primarily addressed in hyper-parameter optimization, with some interesting solutions proposed.

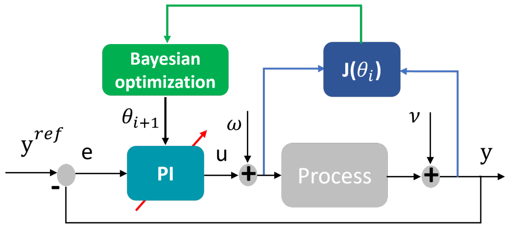



Figure 2: BO-based PI controller auto-tuning, where $i$ indicates a closed-loop experiment with controller parameters $\theta_i$.

For instance, (Perrone et al., 2019) suggested limiting the search space of the optimized process (target task) to sub-regions extracted from historical data. In (Li et al., 2022), a different two-phase transfer learning framework for automatic hyper-parameter optimization is proposed, which can simultaneously handle the complementary nature among source tasks and dynamics during knowledge aggregation. This solution is adopted in this paper to accelerate the Bayesian optimization framework, and the approach will be presented in the next section.

## 3 ACCELERATE THE PI TUNING IN COMMISSIONING PHASE USING TRANSFER LEARNING

In this section, we explain how to use transfer learning technique for accelerating the auto-tuning of a PI controller in the commissioning phase. Without loss of generality, it is assumed that historical data is available or can be generated from existing simulation models. Indeed, data can be generated from variants of the same model by altering the values of unknown or uncertain parameters. For instance, if you have a simulation model with several parameters whose exact values are not known, you can create different scenarios by systematically varying these parameters within plausible ranges. This process allows you to explore a wide range of possible outcomes and generate a diverse datasets. Additionally, this approach provides a richer set of data that captures the variability and uncertainty inherent in real process.

The transfer learning is conducted using the approach proposed by (Li et al., 2022) and illustrated in Fig. 3. This approach addresses two main challenges. The first challenge is how to tackle the complementary nature among source tasks and extract source knowledge cooperatively. To solve this, TransBO builds a source surrogate by combining base surrogates from multiple source tasks using learned weights. This phase ensures that the complementary

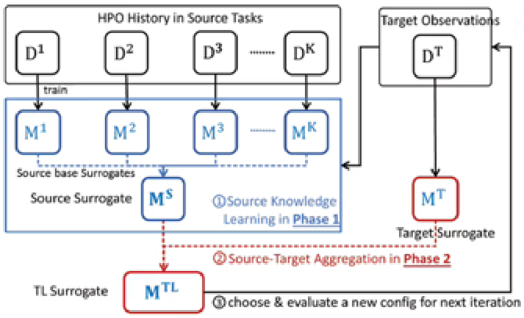



Figure 3: Two-Phase Transfer Learning Framework (Li et al., 2022).

nature of different source tasks is utilized effectively. The second challenge is how to handle knowledge transfer as more target measurements become available. This is solved in a second phase, where the source surrogate is combined with the target surrogate to form the final transfer learning surrogate. This combination is done in an adaptive manner, allowing the framework to balance between source and target knowledge dynamically. TransBO learns the weights for combining surrogates by solving constrained optimization problems with a differentiable ranking loss function. This principled approach ensures that the knowledge transfer is both effective and adaptive, leading to better performance with fewer evaluations. To maximize the generalization ability of the transfer learning surrogate, TransBO uses cross-validation to learn the aggregation weights. This helps in preventing overfitting and ensures that the surrogate model remains robust.

# 4 AUTO-TUNING PI CONTROLLER FOR A THERMAL PLANT

## 4.1 Set-Up

As a practical use case a thermal plant setup has been used to demonstrate and validate the tuning methods in a realistic environment, resembling on one end cooling of powertrain components, and on the other typical industrial plastic curing, and drying applications.

The setup shown in figure 4, consists of an aluminum plate split into five distinct zones each heated by a cartridge heater of 113W, which is in turn powered by a solid-state relay (SSR) applying a PWM 0-220 V DC. Next to the heating, each of the zones is also actively cooled on the back end by liquid cooling through a cooling channel using water-glycol. The valves on the entry of each of the five cooling channels allows for opening and closing, while the speed controllable pump, enables flow control through the opened channels.

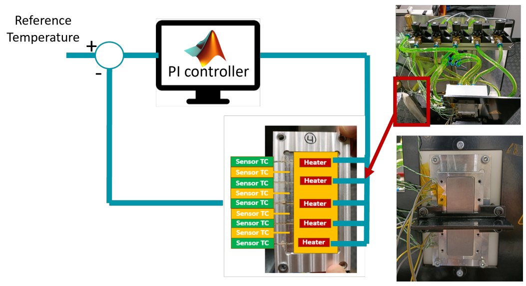



Figure 4: Experimental setup: thermal plant.

The thermal measurements are recorded by a total of nine (9) thermocouples, glued into each of the five aluminum zones, but also the boundary between each two zones. The thermocouples are calibrated to +-1C accuracy. The fluid temperature are measured by Pt100 sensors on input/output coolant lines. Additionally, pressure across the setup and the flow rate of the water glycol are measured, both for safety and for validation purposes.

All the controls and measurements are interfaced to a Beckhoff I/O stack connected over EtherCAT to a Xenomai Triphase real-time target. The base sampling time of the control loop is Ts = 1ms while the sensor measurements are logged at Tmeas = 100ms, which is deemed sufficient for a rather slow thermal process. This allows for rapid prototyping of control code in Simulink and easy integration to test environment in MATLAB.

## 4.2 Thermal Plant Model

As mentioned in 3, a model is needed for calculating the initial condition and the upper and lower bounds of the controller gains before running the BO with the real plant in the commissioning phase. The used model is a physics-based white box model with dynamic of heat dissipation, convection losses and heat exchange between different components. This plant model is excited by multiple disturbances and evaluated in a closed loop simulation with a feedback controller (PI).

### 4.2.1 Thermal Model Assumptions

The thermal plant model is a second order model, consisting of two connected metal masses. The model buildup and numerical evaluation was performed in python. The thermal model under investigation includes the main modes of heat exchange between and within the components:

- One block, "mass 1 (copper)" is heated internally, through direct heat injection. this is equivalent to heating by an electric resistor inserted in this volume.
- A second block, "mass 2 (aluminium)" is cooled internally, driven by a cooling temperature and given transfer area and heat transfer coefficient. This is equivalent to cooling through a glycol-water mixture, circulating at high throughput rate through pipes in the volume).
- The two metal blocks (aluminium and copper) are connected and do exchange (when at different temperatures) heat by conduction through an area of contact.
- Also, both blocks are in contact with surrounding air and lose or gain thermal energy through convection.

#### 4.2.2 Model Structure

The model structure is as follows:

Two states are defined, representing the bulk copper temperature ($T_{node,Cu}$) and the bulk aluminum temperature ($T_{node,Alu}$). Heat injection and loss (through cooling and resistive heating) and heat losses or gains are injected directly into these states.

The node temperatures are updated every time step using a backward Euler scheme, with the combined energy inputs of each of the heat losses and gains.

The heat gains and losses from heating $dT_H$ and cooling $dT_H$ are a defined as:

$$dT_{H,t-1\to t} = Q_H/m/cp * dt$$

and

$$dT_{C,t-1\to t} = Q_C/m/cp * dt$$

whereby the heat gain $Q_H$ is driven directly by the PI-controller, and

$$Q_C = -(h_{alu,C} * A_{alu,C} * U_{alu,C} * (T_{C,t} - T_{alu}) * dt)$$

As the two blocks have a different temperature and are physically connected along area $A$, a heat flow through conduction $dT_{othermat}$ will be induced. The flux $q_{Alu,Cu}$ is a function of their respective temperatures and distance of bulk temperature nodes. The joint temperature is defined as a function of to the perpendicular distances between the cooling and heating nodes and the joint:

$$T_{joint,t} = \frac{T_{cu} * k_{alu} * L_{node,alu} + T_{Alu} * k_{cu} * L_{node,cu}}{k_{alu} * L_{node,alu} + k_{cu} * L_{node,cu}}$$

The resulting heat flow $Q_{Alu,Cu}$ depends on the contact area and mass of the materials(s):

$$\begin{aligned} A_{alu,cu} * q_{Alu,Cu} &= -A_{alu,cu} * q_{Cu,Alu} \\ &= (T_{node,Alu} - T_{joint,t}) * cp_{alu} * m_{alu} * dt \end{aligned}$$

On the other hand, heat loss to the environment is modelled using conduction through a given surface $A_{air}$ with thermal transmittance $U_{air}$ and $T_{ext}$ the exterior temperature:

$$dT_{air,t-1\to t} = -(T_{node,t-1} - T_{ext}) * A_{air} * U_{air}/m/cp * dt$$

The numerical values for the model parameters and initial state are summarized in Table 1 of (Boulkroune et al., 2024). For this analysis, these values are primarily held constant; however, they can be made variable, and noise can be introduced into the disturbance, input, and output signals.

## 4.3 Results and Benchmarking

Experiment description follows (see Figure 5). Each experiment lasts for a total of 380 seconds. It begins with a warmup to 25C for min relying on a well-tuned PI parameters (this ensures consistency in comparison between different runs), followed by a bumpless transfer of control parameters to the tested PI combination, so that the sudden change in integrator parameter does not create a large step in the output of the controller. Finally, a step reference is applied at 140 seconds and is recorded for length of 120 seconds. The experiment ends with a cool-down period of 120 seconds back to below 25C so that the next experiment can be started. For the computation of the objective function, exclusively the data within the time interval of 140 to 260 seconds is considered.

The description of how the experiments were realized is illustrated in Figure 5. Each experiment lasts for a total of 380 seconds. It begins with a warm-up to 25°C for a few minutes, relying on well-tuned PI parameters (this ensures consistency in comparison between different runs). This is followed by a bump less transfer of control parameters to the tested PI combination, so that the sudden change in the integrator parameter does not create a large step in the output of the controller. Finally, a step reference is applied at 140 seconds and is recorded for a length of 120 seconds. The experiment ends with a cool-down period of 120 seconds back to below 25°C so that the next experiment can be started. For the computation of the objective function, exclusively the data within the time interval of 140 to 260 seconds is considered.

In a previous study, we conducted a total of 256 experimental runs based on the procedure explained earlier. Due to the significant time and financial costs

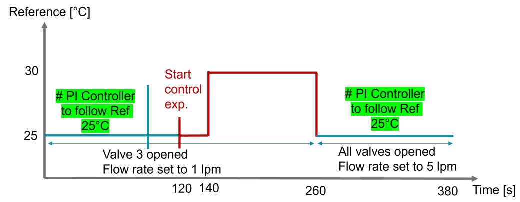



Figure 5: Experiment description.

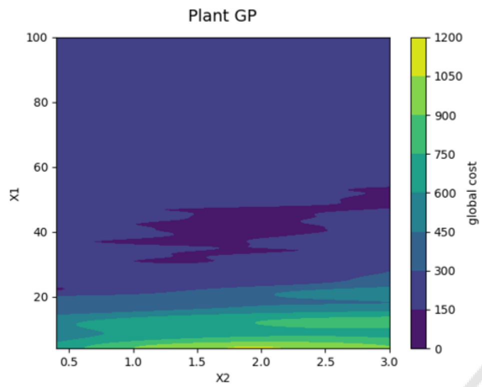
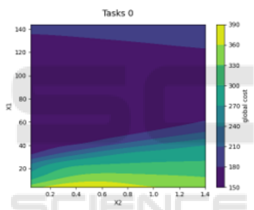
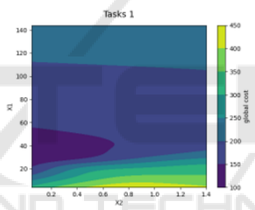


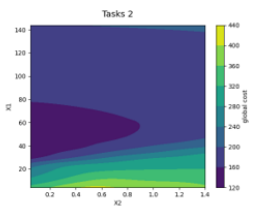

Figure 6: Cost heat map.



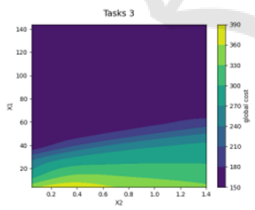

(a) $m_{cu} = 0.5 * \hat{m}_{node,cu}$



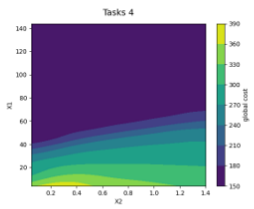

(b) $m_{cu} = 0.75 * \hat{m}_{node,cu}$




(c) $m_{cu} = 1 * \hat{m}_{node,cu}$




(d) $m_{cu} = 1.25 * \hat{m}_{node,cu}$




(e) $m_{cu} = 1.75 * \hat{m}_{node,cu}$

Figure 7: Data generated from different thermal model variants.

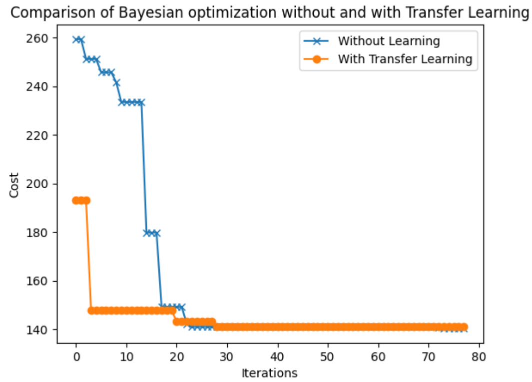



Figure 8: Comparison of BO without and with transfer learning.

associated with repeating these tests, we opted to utilize the collected data by fitting it to a Gaussian Process (GP) model. This model is then used as a surrogate for the actual plant, thereby avoiding the need for further costly and time-consuming experiments. The cost heat map of the collected data is presented in Figure 6. On the other hand, to compensate for the lack of historical data, the presented thermal model will be used for generating data. First, the unknown parameters in the thermal model are identified using a Bayesian approach. The main idea is to enhance the model prediction capability by trying to identify as accurately as possible the unknown parameters. It is both reasonable and non-limiting to assume that the model parameters are identifiable. Alternatively, it may be feasible to downsize the model, retaining only the identifiable sub-system whenever applicable. At this stage, acquiring experimental data is essential to understand the true dynamics of the plant and only one single experiment is required. It is worth noting that the objective is not to get an exceedingly precise model, but rather to achieve a model of satisfactory accuracy, suitable for generating historical data from the model variants. These variants can be obtained from the identified model by varying the values of the identified unknown parameters within a specified ranges. Other alternatives, if you have a simulation model with several parameters whose exact values are not known, you can create different scenarios by systematically varying these parameters within plausible ranges. In our study, the first approach is used. The mass of the cupper (i.e,$m_{node,cu}$) is the only unknown parameters identified and five (5) simulation data are obtained using the values : $0.5 * \hat{m}_{node,cu}$, $0.75 * \hat{m}_{node,cu}$, $1 * \hat{m}_{node,cu}$, $1.25 * \hat{m}_{node,cu}$ and $1.75 * \hat{m}_{node,cu}$, where $\hat{m}_{node,cu}$ is the identified parameter ($\hat{m}_{node,cu} = 0.8262$). The obtained cost heat maps are presented in figure 7. This data will be used as historical data for transfer learning purposes. As we can see, this data exhibits different optimal regions depending on the mass of the cupper.

Instead of conducting a comparison with manual tuning, we opted to employ another optimization approach: Bayesian Optimization (BO) without transfer learning. For both BO techniques (with and without transfer learning), we used the developed tool 'Openbox' by (Jiang et al., 2024), which is publicly available at [a]. For BO without transfer learning, the default configuration was used (i.e, $surrogate\_type =' gp'$, $initial\_runs = 1$, $init\_strategy =' default'$). For BO with transfer learning, the following configuration was used: $initial\_trials = 5$, $init\_strategy =' default'$, $surrogate\_type =' tlbo\_topov3\_gp'$, $acq\_optimizer\_type =' random\_scipy'$. The considered ranges (upper and lower bounds) for the proportional gain (KP) and integral gain (KI) were defined as $[4,100]$ and $[0.1,3]$, respectively. The initial guess is chosen as $K_{p0} = 89.96$ and $K_{i0} = 0.2003$ for both approaches.

The comparison results between BO with and without transfer learning are shown in Figure 8. The x-axis represents the number of iterations, while the y-axis represents the cost. Evidently, the BO with transfer learning achieves a convergence rate that is 76% faster than BO without transfer learning. A notable performance obtained even considering the moderate model prediction quality. There is potential for further enhancement in this percentage if additional efforts are invested in refining the model predictions.To ensure a fair comparison between the two optimization algorithms, it is advisable to run them with several initial guess points. This approach helps to mitigate the influence of any single starting point on the optimization results, providing a more comprehensive evaluation of each algorithm's performance. By using multiple initial guesses, we can better assess the robustness and effectiveness of the algorithms across a broader range of scenarios. Due to time constraints, this will be done in the future, together with direct application to the real thermal plant.

# 5 CONCLUSION

In this study, we proposed a novel approach to accelerate the auto-tuning of PI controllers during the commissioning phase. By combining transfer learning and Bayesian optimization, we aimed to minimize the number of iterations needed to reach the optimal solution. Transfer learning was utilized to extract valuable insights from historical data obtained from a simulation model. The effectiveness of our approach was demonstrated through its application to a thermal plant, significantly reducing the number of iterations required to achieve the optimizer's optimal solution.

[a]https://github.com/PKU-DAIR/open-box

# ACKNOWLEDGMENT

The research was supported by VLAIO under the project HBC.2022.0052 Tupic-ICON performed in Flanders Make.

# REFERENCES

Aidan, O. (2006). *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press.

Bazanella, A. S., Campestrini, L., and Eckhard, D. (2011). *Data-driven controller design: The $h_2$ approach*. Springer Science and Business Media.

Boulkroune, B., Jordens, X., Mrak, B., Verhelst, J., Depraetere, B., Meskens, J., and Bovijn, P. (2024). Enhancing pi tuning in plant commissioning through bayesian optimization. In *ECC 2024*, page 3220–3225.

Boyd, S., Hast, M., and Åström, K. (2016). Mimo pid tuning via iterated lmi restriction. *Int. J. Robust Nonlinear Control*, 26:1718–1731.

Campi, M., Lecchini, A., and Savaresi, S. (2002). Virtual reference feedback tuning: A direct method for the design of feedback controllers. *Automatica*, 38(8):1337–1346.

Doerr, A., Nguyen-Tuong, D., Marco, A., Schaal, S., and Trimpe, S. (2017). Model-based policy search for automatic tuning of multivariate pid controllers. *arXiv*. arXiv:1703.02899.

Fujimoto, Y., Sato, H., and Nagahara, M. (2023). Controller tuning with bayesian optimization and its acceleration: Concept and experimental validation. *Asian J Control*, 25:2408–2414.

Garnett, R. (2023). *Bayesian Optimization*. Cambridge University Press, Cambridge, 1st edition.

Gevers, M. (2002). *Modelling, identification and control*, page 3–16. Springer-Verlag.

Hjalmarsson, H. (1998). Control of nonlinear systems using iterative feedback tuning. In *Proceedings of the 1998 American Control Conference*, page 2083–2087.

Ho, W., Hong, Y., Hansson, A., Hjalmarsson, H., and Deng, J. (2003). Relay autotuning of pid controllers using iterative feedback tuning. *Automatica*, 39(1):149–57.

Jiang, H., Shen, Y., Li, Y., Xu, B., Du, S., Zhang, W., Zhang, C., and Cui, B. (2024). Openbox: A python toolkit for generalized black-box optimization. *Journal of Machine Learning Research*, 25(120):1–11.

Kaneko, O., Soma, S., and Fujii, T. (2005). A new approach to parameter tuning of controllers by using one-shot experimental data - a proposal of fictitious reference iterative tuning. In *16th IFAC World Congress*, volume 38, page 626–631.

Kinoshita, T. and Yamamoto, T. (2018). Design of a data-oriented pid controller for a two degree of freedom control system. In *3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID*, volume 51, page 412–415.

Li, Y., Shen, Y., Jiang, H., Zhang, W., Yang, Z., Zhang, C., and Cui, B. (2022). Transbo: Hyperparameter optimization via two-phase transfer learning. In *SIGKDD 2022*.

Liuping, W. (2017). Automatic tuning of pid controllers using frequency sampling filters. *IET Control Theory Appl*, 11(7):985–95.

Lu, H., Shu-yuan, W., and Sheng-nan, L. (2017). Study and simulation of permanent magnet synchronous motors based on neuron self-adaptive pid. In *2017 Chinese Automation Congress (CAC)*, page 2668–2672, Jinan, China.

Perrone, V., Shen, H., Seeger, M., Archambeau, C., and Jenatton, R. (2019). Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. In *Advances in Neural Information Processing Systems*, volume 32.

Reynoso-Meza, G., Sanchez, H. S., Blasco, X., and Vilanova, R. (2014). Reliability based multiobjective optimization design procedure for pi controller tuning. In *19th World Congress*, page 10263–10268.

Sanchis, R. and Peñarrocha-Alós, I. (2022). A new method for experimental tuning of pi controllers based on the step response. *ISA Trans.*, 128:329–342.

Shipman, W. (2021). Learning to tune a class of controllers with deep reinforcement learning. *Minerals*, 11(9):989.