Metamorphic Testing for Optimisation: A Case Study on PID Controller Tuning

Alejandra Duque-Torres[a], Claus Klammer[a], Stefan Fischer[a], Rudolf Ramler[a], Dietmar Pfahl[b]

[a]*Software Competence Center Hagenberg (SCCH) GmbH, Softwarepark 32a, Hagenberg Im Mühlkreis, 4232, Upper Austria, Austria*
[b]*Institute of Computer Science, University of Tartu, Narva mnt 18, Tartu, 51009, Tartu, Estonia*

Abstract

Context: Tuning PID controller parameters is essential for achieving stable and efficient system behaviour. However, traditional methods often depend on expert heuristics or exhaustive simulations, which can be time-consuming and difficult to generalise across configurations.

Objective: This paper explores Metamorphic Testing (MT) as a validation-driven approach for identifying optimal PID parameter configurations based on expected behavioural trends.

Method: We define a set of domain-informed Metamorphic Relations (MRs) that capture control-theoretic expectations. These MRs are used to evaluate and compare simulated parameter configurations, based on how many behavioural constraints they satisfy.

Results: Our findings show that high MR satisfaction is a strong indicator of potentially optimal parameter configurations, often yielding smooth, stable control responses. Although trade-offs between metrics—such as energy usage and error correction—occasionally led to partial MR violations, expert-preferred parameters set still closely matched those with the highest MR scores. These results support the use of MR-based evaluation as a practical approach to identifying high-quality tuning parameter.

Conclusion: Metamorphic Testing offers an interpretable and structured way to guide controller tuning by formalising expected system behaviour. This approach helps identify near-optimal configurations and complements traditional tuning practices through systematic validation.

Keywords: Metamorphic Testing, Metamorphic Relations, Search, PID controller, Parameter Tuning

1. Introduction

Optimisation refers to the process of finding the most effective solution to a given problem based on a measure of "goodness". A common optimisation challenge is the search and selection of key system parameters to ensure the desired system behaviour [1]. In control systems, for instance, fine-tuning the operational parameters of controllers is essential for balancing stability, improving response time, and enhancing efficiency while reducing cost [2, 3, 4]. Similarly, in machine learning, optimising hyper-parameters such as the learning rate, batch size, and network depth plays a crucial role in improving model performance [5, 6].

Several approaches have been introduced to tackle this problem, including empirical methods [7], heuristic strategies [8], mathematical optimisation [8], and automated search techniques [9]. However, the effectiveness of any parameter search method depends on how well the selected parameters perform in practice [10, 11]. Thus, robust verification plays a crucial role in determining whether a given parameter set truly achieves the desired system behaviour [12]. Traditional verification techniques rely on evaluating system performance against predefined benchmarks [13], such as efficiency, accuracy, stability, or cost-effectiveness, depending on the domain. In some cases, simulation-based testing allows parameters to be assessed in controlled virtual environments before deployment [14].

In complex systems where direct verification is challenging, methods like sensitivity analysis [13], robustness testing [15], and uncertainty quantification [16] help assess how parameters influence system behaviour under different conditions. Yet, these verification techniques often require explicit expectations or reference outputs, which may be unavailable or infeasible to define—particularly when assessing numerous parameter configurations or exploring new operating conditions. Metamorphic Testing (MT) has been proposed as an alternative verification strategy in such contexts, especially where the absence of a test oracle complicates traditional verification [17]. Rather than relying on absolute correctness, MT verifies system behaviour by checking whether expected relations between inputs and outputs—captured as Metamorphic Relations (MRs)—are preserved across variations in the inputs [18].

While MT is typically applied as a verification strategy, in this paper, we explore its potential not only to verify system behaviours but also to guide the search for suitable parameter configurations. Our hypothesis is that, by leveraging domain-specific expectations encoded as MRs, MT can help determine whether alternative configurations preserve the desired behaviour and perform at least as well as the reference. In addition, parameter configurations that satisfy the most MRs are more

likely to be strong candidates for optimal tuning, as they align with expected system behaviour and demonstrate consistent—or even improved—performance.

This work is motivated by our collaboration with an industry partner in the field of machinery and plant engineering. Their machines are designed to support a wide range of production scenarios, with varying sizes, power levels, and operational requirements. One of the core components for controlling physical processes in these machines is the PID (Proportional–Integral–Derivative) controller, which adjusts control signals based on the error between desired and actual values. The controller's performance depends on the correct configuration of three key parameters—proportional, integral, and derivative gains—that must be tuned to achieve optimal behaviour under specific operating conditions. However, because the precise application of each machine is often unknown to the manufacturer, customers are typically responsible for configuring and optimising the controllers to meet their specific needs. An initial set of PID parameters is usually provided based on expert knowledge, simulations, and prior experience. Yet, as system conditions evolve—due to mechanical modifications, environmental changes, or shifting operational demands—this initial configuration may no longer ensure optimal performance. This creates the need for a structured approach to identifying alternative parameter sets that preserve the expected behaviour of the PID controller while adapting to new conditions. In this context, we explored the use of MT to support the evaluation and selection of promising parameter configurations.

In this paper, we report on our experience applying MT to guide the search for improved controller settings in a realistic industrial scenario. The remainder of the paper is structured as follows. Section 2 presents the background, including the main concepts used in this study as well as related work. In Section 3, we introduce the industrial context. Section 4 describes the process we followed to support the evaluation and selection of promising parameter configurations, including implementation details, the MRs used, and a description of the system under test (SUT). Section 5 presents the results of our study, and Section 6 provides a discussion of the findings. Finally, Section 7 concludes the paper.

2. Background and Related Work

This section presents the key concepts utilised in our study, as well as the related work. Section 2.1 introduces the MT approach. Section 2.2 provides an overview of PID controllers and common tuning methods. Section 2.3 defines the key controller parameters and performance metrics used throughout this study. We also present the equations used to compute the standard PID gains based on the normalised tuning

parameters. Finally, Section 2.4 reviews related work on MT, with a particular focus on its application in the context of PID controller tuning.

2.1. Metamorphic Testing

Metamorphic Testing (MT) is a software testing technique introduced by Chen et al.[17] to address the test oracle problem, *i.e.,* the challenge of determining the correctness of outputs when no reliable oracle is available or to expensive to develop. Instead of focusing on absolute correctness, MT examines whether the SUT behaves consistently with a set of expected relations between inputs and outputs—known as Metamorphic Relations (MRs). These MRs express how changes in the input should affect the output across multiple executions of the SUT [18].

The typical MT process consists of the following steps:

1. Generate an initial set of test data (TD).
2. Identify applicable MRs for the SUT.
3. Transform the original test data into new test data (transformed test data, TTD) according to the selected MRs.
4. Execute both the original TD and the corresponding TTD on the SUT.
5. Verify whether the observed outputs conform to the expectations defined by the applied MRs.

While MT can uncover faults when an MR is violated, the absence of violations does not imply correctness. Final interpretation depends on the soundness and applicability of the selected MRs to the given input domain [19].

2.2. Basics of PID Controllers and Tuning Methods

Control systems are fundamental to a wide range of applications, including industrial automation, robotics, and aerospace engineering [3, 2, 20, 4]. Among the various control strategies, the Proportional–Integral–Derivative (PID) controller remains one of the most widely used due to its balance of simplicity and effectiveness [21]. A PID controller continuously adjusts control signals based on the error between the desired set-point and the actual process variable. This adjustment is computed using three components: proportional (K_p), integral (K_i), and derivative (K_d) gains [21], Figure 1 provide an overview. These components enable the controller to respond proportionally to the current error, correct for accumulated past errors, and anticipate future trends based on the error's rate of change. This mechanism makes PID controllers essential for maintaining system stability and achieving precise control in dynamic environments [22].

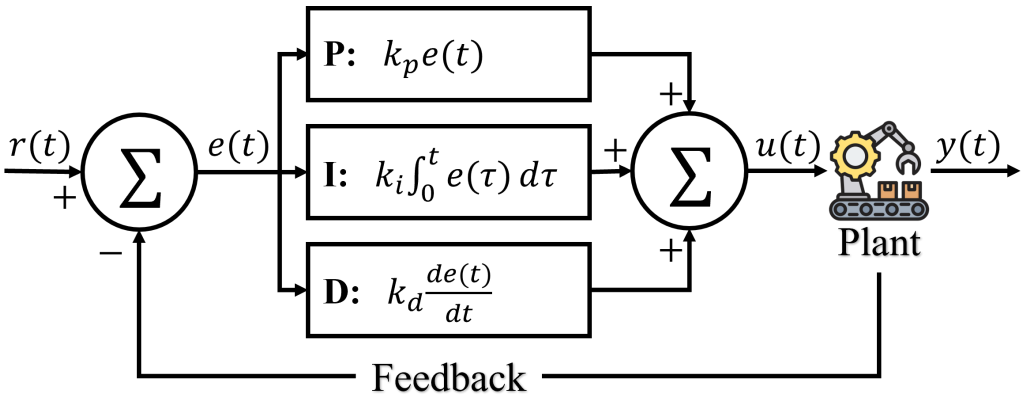

Figure 1: Diagram of a PID (Proportional–Integral–Derivative) control loop. The controller calculates the control signal as the sum of three terms: proportional to the error $e(t)$, integral of the error over time, and derivative of the error. These components are weighted by the parameters K_p, K_i, and K_d, which must be tuned to achieve the desired system response.

The performance of a PID controller strongly depends on the proper tuning of its parameters. Poorly tuned parameters can lead to performance issues such as overshoot, instability, oscillations, or slow settling times. Effective tuning aims to strike a balance among these factors to ensure robustness and responsiveness under varying conditions. Several methods have been proposed for PID tuning, which can be broadly classified into four categories: classical methods, data-driven methods, optimisation-based methods, and adaptive or self-tuning methods. Classical methods include empirical and analytical techniques such as Ziegler–Nichols [23] and Cohen–Coon [24], which offer heuristic rules based on system response. They are simple to apply but may not yield optimal results for complex or non-linear systems [25].

Data-driven approaches use historical or simulation data to adjust parameters without requiring an explicit model. Examples include Virtual Reference Feedback Tuning (VRFT) [26] and Fictitious Reference Iterative Tuning (FRIT) [27]. These methods are particularly effective for non-linear or time-varying systems [7]. The optimisation-based methods employ computational algorithms to explore the parameter space and minimise performance-related cost functions. Common approaches include Genetic Algorithms [28] and Particle Swarm Optimisation [9], which iteratively refine the controller parameters [9]. Adaptive and self-tuning methods dynamically adjust parameters in real time to respond to changing system dynamics. Techniques such as Model-Reference Adaptive Control (MRAC) [29] and Self-Tuning Regulators [30] fall into this category and aim to improve controller robustness in uncertain or evolving environments [8].

2.2.1. Real-World PID Tuning in Industrial Applications

While numerous PID tuning methods exist in theory, their practical application in industrial environments often deviates due to constraints such as hardware limitations, system variability, or real-time performance requirements. In practice, tuning typically involves a combination of automated tools and manual intervention. Many industrial systems—such as those based on Programmable Logic Controllers (PLCs) or Distributed Control Systems (DCS)—integrate built-in auto-tuning functions. Commercial platforms like MATLAB Simulink [31], Siemens TIA Portal [32], and Rockwell Studio 5000 [33] offer PID auto-tuners that analyse system responses and automatically suggest parameter values. These tools often rely on techniques such as step response analysis, relay feedback, or gain–phase margin optimisation. However, most auto-tuning methods assume linear system behaviour and may not perform well for non-linear or time-varying processes [30].

Despite the availability of such tools, manual tuning remains widespread in industry. Engineers frequently rely on heuristic knowledge, visual inspection of the system response, or iterative trial-and-error to adjust controller parameters. This approach is particularly valuable in applications such as hydraulic actuators, large-scale manufacturing machinery, or high-precision motion control, where empirical expertise plays a critical role. Nevertheless, manual tuning is time-consuming, less reproducible, and heavily dependent on the operator's experience and domain knowledge.

2.3. Definitions and Performance Criteria

Some PID controllers use normalised parameters instead of the conventional PID gains:

- K_r (normalised Gain): The base gain applied to the error signal.
- T_n (Reset Time): Also called the integral time, it defines the duration over which past errors are accumulated.
- T_v (Rate Time): Also known as the derivative time, it determines the anticipation window for predicting future error trends.

Using these normalised parameters, the conventional PID gains are calculated as:

$$K_p = \frac{1}{K_r} \tag{1}$$

$$K_i = \frac{K_r}{T_n} \tag{2}$$

$$K_d = K_r \cdot T_v \tag{3}$$

The following metrics are used to evaluate the performance of each PID configuration:

- **Overshoot (%OS):** The percentage by which the system output exceeds the target value during its transient response. It is computed as:

  $$\text{Overshoot} = \frac{y_{\text{peak}} - y_{\text{target}}}{y_{\text{target}}} \cdot 100$$

  In control system design, an overshoot of up to 5% is commonly considered acceptable, as higher values may lead to undesirable instability or excessive actuator effort.

- **Rise Time (t_r):** The time required for the system output to rise from 10% to 90% of the target value. It reflects how quickly the system reacts to a step change.

- **Settling Time (t_s):** The time required for the system output to remain within a specified range (typically $\pm 2\%$ or $\pm 5\%$) of the target value following a disturbance or set-point change. It reflects how quickly the system stabilizes after a transient response.

- **Steady-State Error (ESS):** The final error between the system output and the reference set-point once the system has settled:

  $$\text{ESS} = \lim_{t \to \infty} |e(t)|$$

- **IAE (Integral of Absolute Error):**

  $$\text{IAE} = \int_0^T |e(t)|\, dt$$

  It sums the absolute tracking error over time.

- **ITAE (Integral of Time-weighted Absolute Error):**

$$\text{ITAE} = \int_0^T t \cdot |e(t)|\, dt$$

  It penalizes errors more strongly as they persist over time.

- **Energy:** The energy of the output or control signal, commonly defined as:

$$E = \int_0^T y^2(t)\, dt$$

  It captures the overall effort of the system response.

- **Oscillations:** Quantified using the number or magnitude of zero-crossings, or via spectral analysis. In our case, we estimate oscillatory behaviour using statistical or signal-based measures over a defined time window.

2.4. Related Work

MT has emerged to address the test oracle problem, especially in contexts where it is difficult or infeasible to define the expected output for every possible input [17, 34]. While MT has been widely adopted in traditional software testing, recent efforts have explored its application in non-traditional domains such as embedded systems, simulations, and cyber-physical systems [35, 36].

Specific studies have investigated the application of MT to control systems, including Proportional–Integral–Derivative (PID) controllers. For instance, Chen et al. [37] applied MT to validate the functionality of a software-based PID controller. Their work defined a set of MRs based on known control-theoretic expectations—such as how output behaviour should change in response to variations in controller gains—and used these to detect implementation faults. Similarly, Kuo et al. [38] proposed a metamorphic testing strategy to verify the consistency of output responses for PID controllers under expected signal transformations. These studies focus primarily on validating PID controller implementations to detect software bugs or deviations from specification.

However, to the best of our knowledge, prior work has not explored the use of MT as a means of evaluating PID parameter configurations during controller tuning. Conventional tuning approaches rely on empirical methods, optimization algorithms, or simulation-based heuristics [3, 25, 8]. While effective, these methods often lack interpretability and may fail to capture domain expectations in a reusable way. In

contrast, MT offers a lightweight, behaviour-driven framework for validating system responses based on domain-informed expectations.

Our work builds on the insights from previous studies but shifts the focus from implementation-level verification to behavioural evaluation and parameter selection. We apply MT retrospectively to simulation data, using domain-specific MRs to evaluate whether a given PID configuration produces system behaviour that aligns with known trends. This allows us to identify parameter sets that are not only valid but also a potential optimal new parameter set.

3. Industry Context

This work has been carried out in collaboration with a long-term industry partner in the field of machinery and plant engineering. The company develops highly configurable machines capable of producing and handling a wide range of products across different domains, depending on customer needs. A large number of machine variants and configuration options are available, specifying characteristics such as size, power, features, and operational flexibility.

This collaboration is part of a broader effort to address the challenges associated with tuning PID controllers in complex and flexible industrial systems. These machines are designed to support diverse production scenarios and must adapt to varying operating conditions. However, despite their high versatility, the precise applications are often unknown at the time of manufacturing. As a result, customers are typically responsible for configuring and optimising the machines—including PID controller settings—to suit their specific use cases.

At the core of this flexibility lies a mechatronic design that integrates mechanical, electronic, and software components. A critical element of this control stack is the PID controller, which regulates key variables such as pressure and speed of hydraulic components. Ensuring robust and optimal performance across a wide range of configurations is a non-trivial task: tuning the controller for one setup must not degrade performance in others. Additionally, evaluating the behaviour of different parameter configurations across many test runs is both time-consuming and highly dependent on expert judgement, making it difficult to scale or standardise the tuning process.

Motivated by this challenge, we engaged with our industry partner to explore the use of MT for guiding and validating PID parameter tuning. Rather than relying on manual trial-and-error or static threshold-based validation, our approach encodes expected behavioural trends as MRs—domain-informed patterns that describe how system behaviour should change as parameters are adjusted. By systematically applying MRs, we can evaluate whether different parameter sets produce behaviour consistent with domain expectations.

We hypothesise that configurations satisfying a larger number of MRs are strong candidates for optimal tuning, as they reflect stable and predictable behaviour. This approach allows us to support tuning decisions with interpretable, data-driven insights—ultimately improving controller adaptability and reducing the manual effort required for configuration validation.

4. Methodology

This section presents the experimental setup and methodology used to investigate the application of MT for evaluating and guiding the selection of PID controller parameters. We begin by describing the SUT in Section 4.1, followed by the experimental procedure in Section 4.2, which includes test data generation, simulation execution, feature extraction, and the application of MT.

4.1. System Under Test

As described in Section 3, this study was conducted in collaboration with an industrial partner specialising in machinery and plant engineering. The SUT is a black-box PID controller responsible for regulating pressure (P) and speed (S) of hydraulic components in a highly configurable industrial machine.

Although the internal configuration of the physical machine is not accessible, the simulations are executed using the company's in-house Simulink-based simulator, which, according to the industrial partner, closely reflects the real machine behaviour. For this study, three representative machine configurations, $\texttt{tc}_{100}$, $\texttt{tc}_{104}$, and $\texttt{tc}_{106}$, were provided. All use the same PID controller and the same initial parameter set, defined by domain experts to represent nominal operating conditions in production systems.

The PID controller operates based on three normalised parameters K_r, T_n, and T_v. These parameters influence the system's response, particularly metrics such as t_r, t_s, OS, and overall stability of both the P and S signals. Its output includes:

- The actual pressure signal over time.
- The actual speed signal over time.
- The reference target signal for pressure (P_{Ref}).
- The reference target signal for speed (S_{Ref}).

Figure 2 illustrates the reference behaviour of the system under the baseline PID configuration for the three machine variants. While configurations $\texttt{tc}_{100}$ and $\texttt{tc}_{104}$ exhibit smooth and stable responses with minimal overshoot and fast settling, $\texttt{tc}_{106}$ shows a significantly different dynamic, including pronounced overshoot and oscillations in both pressure and speed.

4.2. Experimental Procedure

The experimental procedure is structured into fourth steps (1) defining an initial parameter set and generating test data by varying these parameters, (2) simulating each parameter set generated, (3) extracting relevant features from the simulation results, and (4) applying the MT process.

4.2.1. Step 1: Defining the Initial Parameter Set and Generating Test Data

The baseline PID parameter set is defined by domain experts and reflects nominal operating conditions used in the current deployed systems. This set includes three normalised tuning parameters K_r, T_n, and T_v.

In this study, test data refers to the set of PID parameter configurations to be evaluated. Each parameter set is defined by a triplet of these normalised parameters. To systematically explore the parameter space, we adopt a sampling strategy based on the Student's t-distribution. For each parameter, values are sampled around the baseline to favour nearby configurations while still allowing the inclusion of more extreme cases due to the distribution's heavier tails.

Sampled values are clipped to remain within valid operational ranges, rounded to reflect the controller's precision, and then combined into unique parameter configurations. This strategy produces a realistic and diverse set of tuning configurations.

4.2.2. Step 2: Simulation Process

Each PID parameter is evaluated using the company's proprietary Simulink-based simulation environment. The system is subjected to a unit step input, and the controller's responses for pressure and speed are recorded over time. For each test case, the following data is collected:

- Test identifier
- Pressure and speed response signals.
- Corresponding reference (target) signals.
- The specific PID parameters used.

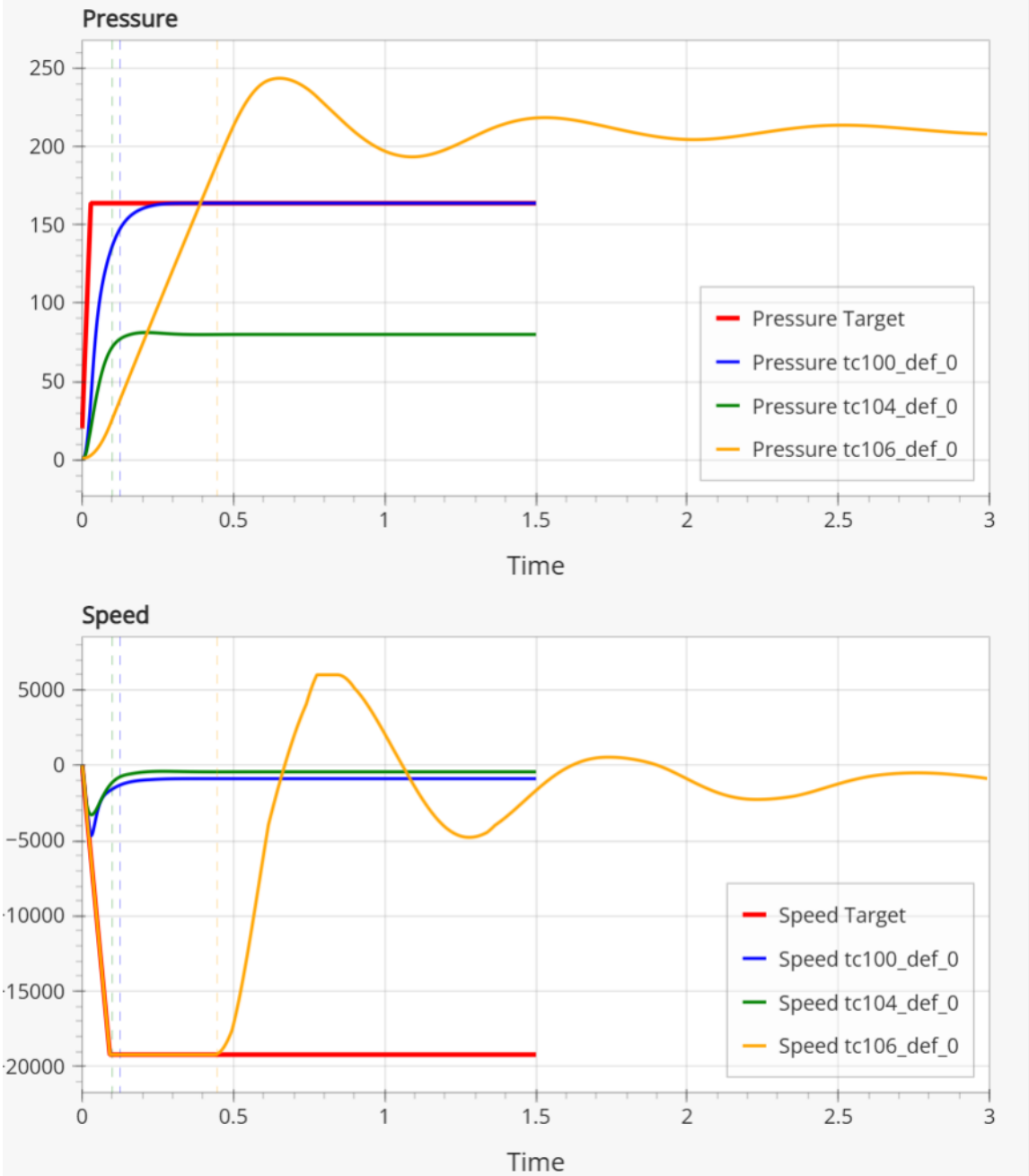

Figure 2: Reference system behaviour for configurations $\texttt{tc}_{100}$, $\texttt{tc}_{104}$, and $\texttt{tc}_{106}$ using the same baseline PID parameters. The top plot shows the pressure responses; the bottom plot shows the corresponding speed responses.

The simulation duration and sampling frequency are kept constant across all parameter configurations to ensure comparability per each configurations, *i.e.,* $\texttt{tc}_{100}$, $\texttt{tc}_{104}$ and $\texttt{tc}_{106}$.

4.2.3. Step 3: Feature Extraction and Preprocessing

To enable analysis and MT evaluation, each simulation output is post-processed to extract a set of quantitative performance metrics. These include:

- PID gains (K_p, K_i, K_d) derived from the normalised parameters.
- Time-domain metrics: OS, t_r, t_s, and ESS.
- Error-based metrics: IEA and $ITAE$.
- Energy metrics: pressure energy (Eng_p), speed energy (Eng_s), and their ratio ($Eng_{p/s}$),
- Signal behaviour indicators: oscillation estimates for both pressure and speed.

4.2.4. Step 4: Applying the Metamorphic Testing Process

Using the extracted features, we apply a set of domain-specific MRs that encode expected behavioural trends. For instance, increasing K_r while keeping T_n and T_v constant is expected to reduce t_r but may increase OS. The MT process consists of:

- Identifying applicable MRs based on the parameter variations,
- Comparing the behaviour of each parameter against the reference case,
- Checking for satisfaction of the expected behavioural trends.
- Analysing the results; Parameter sets that satisfy the highest number of MRs are considered more likely to be potentially optimal, as they align with expected behavioural trends derived from control theory.

Each MR can be classified into one of three possible states based on the observed outcome:

- **Satisfied**: The system's behaviour changes in accordance with the expected trend encoded in the MR.
- **Not satisfied**: The system's behaviour contradicts the expected trend, indicating a potential performance issue or suboptimal configuration.

5. Results

A replication package containing the complete dataset generated during our experiments, along with all scripts used for analysis, is available in our public GitHub repository[1]. This section presents the main findings from our evaluation, focusing on the effectiveness of the MT-based approach in identifying promising PID configurations across the three machine variants.

5.1. Step 1: Defining the Initial Parameter Set and Generating Test Data

This initial step focused on establishing the reference PID configuration that serves as the starting point for our analysis. The reference parameter set was not developed as part of this study; rather, it had already been defined and used by the industry partner in real production environments over an extended period. It represents the baseline configuration currently deployed across various machine variants.

To support our study, several technical workshops were held in collaboration with the company. During these sessions, domain experts provided the necessary context about the controller setup, Simulink implementation, parameter conventions, and typical performance characteristics expected from the system. Furthermore, the company facilitated access to the code repository containing the Simulink simulation code, the PID controller software, and corresponding example setups for simulation test runs, enabling us to evaluate new PID configurations in a controlled and reproducible manner.

Due to confidentiality agreements, we are unable to share the Simulink models or internal details of the PID implementation. However, the generated parameter sets and the results of all simulations conducted as part of this study are made publicly available. The corresponding values for the reference configuration, along with selected performance metrics for each machine variant, are summarised in Table 1. In addition to the previously defined performance metrics, we introduce the Pressure Ratio (PR), which is defined as the ratio between the peak pressure value and the target pressure value:

$$PR = \frac{P_{\text{peak}}}{P_{\text{target}}}$$

A PR value close to 1 indicates that the system closely reaches the target without significant overshoot. Values significantly greater than 1 suggest overshooting, while values below 1 indicate that the target was not reached.

[1] `https://github.com/aduquet/SpecialIssue-MT-2025`

Table 1: Reference PID Parameters $K_r = 0.09$, $T_n = 0.008$, $T_v = 0.05$ and Performance Metrics for Each Machine Variant

Config	K_p	K_i	K_d	**PR**	t_r	**P-Eng**	**S-Eng**	**ESS**	**IAE**	**ITAE**
tc_{100}	11.1111	1388.8889	0.5556	1.000	0.126	2.54e8	9.91e11	1.754e-05	7.9761	0.4690
tc_{104}	11.1111	1388.8889	0.5556	1.015	0.100	6.26e7	5.98e11	6.715e-06	4.0463	0.1870
tc_{106}	11.1111	1388.8889	0.5556	1.160	0.446	1.14e8	6.72e12	0.9174	0.1	0.0053

To generate a realistic and diverse set of PID parameters, we developed a set of Python scripts and employed the combinatorial test data generation tool PICT[2] to automate the sampling and preprocessing pipeline. The core script employs the Student's $t-$distribution to generate values for each parameter (K_r, T_n, and T_v) around a specified baseline. This distribution allows for a higher concentration of values near the reference point, while still capturing extreme configurations due to its heavier tails. Each generated sample is clipped to remain within predefined operational limits, rounded based on the system's resolution, and saved to a shared file in a structured format.

The sampling process is controlled via command-line parameters, specifying the value bounds, resolution (step size), and the desired number of samples. The script also generates distribution plots for visual inspection of the resulting sample density. To facilitate downstream processing and traceability, an additional script converts the generated samples into `JSON` format, assigning each test case a unique identifier and transforming variable names into a format compatible with Matlab/Simulink tooling. Figure 3 illustrates the sampling distributions used for K_r, T_n, and T_v.

Table 2 provides an overview of the sampling configuration for the three PID parameters. It lists the default, minimum, and maximum values, the resolution step size, and the number of samples actually generated per parameter. The final column also shows the percentage of the theoretical maximum coverage, highlighting how the sampling is concentrated near the baseline while still maintaining a reasonable spread across the valid range.

5.2. Step 2: Simulation Process

A total of 1000 unique PID parameter configurations were generated from Step 1 and used as input to the controller within the simulation environment. These simulations were executed using the industrial partner's proprietary Simulink-based model, which is designed to emulate the behaviour of the physical machine with high

[2] `https://github.com/microsoft/pict`

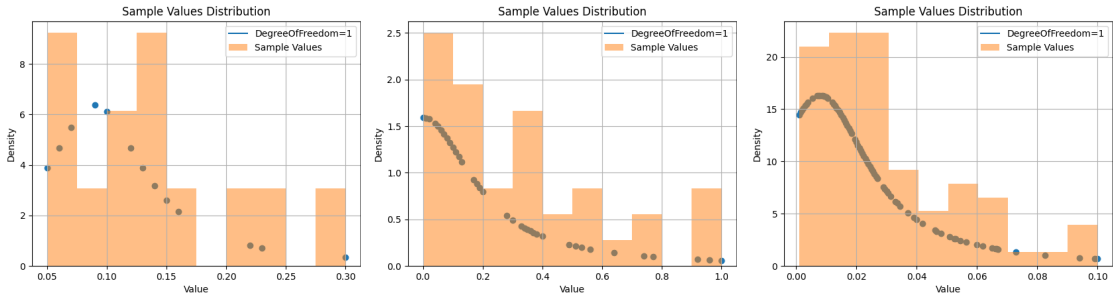

Figure 3: Sampling distributions for PID parameters used to generate test configurations. Each distribution is centred around the baseline parameter using a Student's t-distribution with degree of freedom = 1. The approach favours values near the baseline while allowing broader exploration. Left: K_r. Middle: T_v. Right: T_n.

Table 2: Sampling configuration and statistics for each PID parameter.

Param	Default	Min	Max	Step	Max Steps	Sample Size	Coverage
K_r	0.09	0.05	0.30	0.01	25	13	52%
T_v	0.05	0.00	0.08	0.005	16	9	56%
T_n	0.008	0.004	0.10	0.0005	192	77	40%

fidelity.

For each configuration, the parameters were injected into the PID controller, which was then subjected to a unit step input. The controller's response in terms of pressure and speed was recorded over time. The simulation process was fully automated, allowing us to systematically test all configurations under consistent conditions across three machine variants, $\texttt{tc}_{100}$, $\texttt{tc}_{104}$, and $\texttt{tc}_{106}$, resulting in a total of 3,003 simulation runs.

5.3. Step 3: Feature Extraction and Preprocessing

Each simulation output was post-processed using a custom Python pipeline to extract key performance metrics. These include:

- **PID gains:** The values K_p, K_i, and K_d were computed from the normalised parameters K_r, T_n, and T_v using the standard relations defined in Section 2.3.
- **Time-domain metrics:** t_r was computed as the time required for the system response to reach the 90% of the target value, while OS was computed as the ratio between the peak value and the reference set-point, the allow overshoot was set to the 5% of the target value, this is a requirement of our industry

partner. t_s and ESS were derived by comparing final values with the reference signal.

- **Error-based metrics:** IAE and $ITAE$ were calculated using the trapezoidal rule, based on the absolute difference between the actual and target signals.
- **Energy metrics:** Energy consumption for both pressure and speed signals was approximated using the squared derivative of the response signal over time. The energy ratio $Eng_{p/s}$ was used to assess control effort balance.
- **Signal behaviour:** Oscillatory behaviour in pressure and speed responses was assessed through frequency estimation based on zero-crossings and peak detection technique.

All signal responses were parsed from the simulation output and aligned using a common time vector. The metrics were then aggregated into a `JSON`. The scripts used for these computations are openly available for reproducibility.

5.4. Step 4: Applying the Metamorphic Testing Process

To assess the behaviour of each parameter configuration, we applied a structured MT process informed by domain-specific expectations encoded as MRs. Prior to the MR evaluation, we analysed how each test parameters differed from the reference (baseline) PID setup. This was done by comparing the computed classical gains K_p, K_i, and K_d across all parameters, which allowed us to group the test data based on relative parameter changes (*e.g.,* all gains higher, some lower, etc.).

Although the controller itself operates using the parameters K_r, T_n, and T_v, we relied on their equivalent classical forms for the analysis. This choice was driven by the fact that K_p, K_i, and K_d offer a more intuitive understanding of how tuning affects system dynamics. Moreover, expressing the MRs in terms of these classical gains allowed us to more clearly define expected behavioural trends, such as the impact of increasing proportional gain on rise time or overshoot.

To further understand how the sampled PID parameters diverged from the reference setting, we encoded the relative change of each gain as a three-letter combination: '`L`' (lower), '`H`' (higher), or '`E`' (equal). This yielded compact descriptors like '`LLL`' (all gains lower than the reference), '`HLH`' (K_p higher than the reference, K_i lower than the reference, and K_d higher than the reference), or '`EEL`'. Table 3 summarises the distribution of these patterns across the test dataset. Notably, the most frequent category is '`LLL`', with 601 occurrences, reflecting the effect of the sampling strategy's emphasis on values near the baseline.

Table 3: Distribution of gain change combinations grouped by category. Each combination uses the symbols `L` (lower), `H` (higher), and `E` (equal) to indicate how the gains differ from the baseline configuration. The first, second, and third positions in each combination correspond to changes in K_p, K_i, and K_d, respectively.

Mostly decreased		**Mostly increased**		**Mixed changes**	
Combination	**Count**	**Combination**	**Count**	**Combination**	**Count**
LLL	601	HHH	23	ELH	21
HLL	86	HLH	92	EHL	6
LHL	52	HHL	29		
ELL	45	EHH	4		
LLH	33	LHH	7		
EEL	1	HEH	1		
Total:	817	**Total:**	156	**Total:**	27

As part of our results, we developed and applied a set of MRs that were formulated after inspecting the test data. These MRs were not predefined in the methodology but emerged during our exploration of how different PID gain combinations varied with respect to the reference parameters. Each MR reflects an expected system behaviour corresponding to a specific pattern of parameter change.

Formally, each MR consists of two parts: (i) an input transformation and (ii) an expected output change. In our context, the input transformation is given by the directional variation of the PID parameters—K_p, K_i, and K_d—which we previously clustered into gain patterns. The expected output is expressed in terms of trends observed in key performance metrics: OS, t_r, Energy Ratio (E_p/E_s), ess, IAE, $ITAE$, and Oscillations.

To guide the formulation of these MRs, we first summarised the expected trends for each gain pattern based on control-theoretic reasoning and domain knowledge. Table 4 presents the expected output changes for each metric across the observed gain combinations, along with brief justifications for these trends.

These expected trends serve as the basis for the second part of each MR—the specific output checks used to assess whether a system response aligns with the anticipated behaviour. Each of these checks is implemented as Boolean conditions, returning 1 if satisfied and 0 otherwise.

Below, we detail the individual MRs, each corresponding to a particular aspect of the expected system behaviour:

Table 4: Expected trends and rationale per gain combination. 'L' (lower), 'H' (higher), 'E' (equal). Metrics include OS, t_r, Energy Ratio (E_p/E_s), ESS, IAE, $ITAE$, and Oscillations.

Comb	**Expected output change**							**Reasoning**
	OS	t_r	**Rat. Eng**	**ESS**	**IAE**	**ITAE**	**Osc**	
`LLL`	↓	↑	↓	Amb*	↑	Amb*	≈	Conservative setup yields slow response and higher error; minimal energy.
`HHH`	≈ or ↓	↓	↑	↓	↓	↓	≈ or ↓	Aggressive gains reduce rise time and ess; K_d damps overshoot; increased effort.
`HLH`	↑	↓	↑	↓	Amb*	Amb*	Amb*	High K_p accelerates response; K_d damps, but low K_i limits accuracy.
`HLL`	↑	↓	↑	↑	↑	↑	Amb*	Unbalanced gains create fast, inaccurate, oscillatory behaviour.
`LHL`, `EHL`	↓	↑	↓	↓	↑	↑	Amb*	Integrator dominates; improves accuracy with sluggish response.
`LLH`, `ELL`, `EEL`, `ELH`	≈ *or* ↓	≈ *or* ↑	≈ *or* ↓	↑ *or* ≈	↑	↑	≈ *or* ↓	Low proportional/integral with strong derivative smoothing effect.
`LEL`	↓	↑	↓	↓	↓	↑	Amb*	Slow but precise; only proportional gain lowered.
`LHH`	≈ *or* ↓	↑	≈ *or* ↑	≈ *or* ↓	↓	≈ *or* ↑	Amb*	Lower K_p slows the initial response. Higher K_i reduces steady-state error, while higher K_d contributes damping. Energy use and oscillations depend on the balance between sluggish correction and added damping.
`EHH`, `HEH`	≈	↓	↑	↓	↓	↓	Amb*	High K_i and K_d yield accurate and smooth response.

Comb: Combination, **Rat. Eng:** Ratio Energy, **Amb*:** Ambiguous, ≈: Approximate equal

Table 5 summarises the input change (combination of the parameter set) and the specific expected output.

- $\texttt{MR}_{PR}$: The ratio of the pressure peak to the target should remain within a 5% tolerance.

$$\texttt{1.0} \leq PR \leq \texttt{1.05}$$

- $\texttt{MR}_{RT}$: The rise time should be faster than or equal to that of the reference.

$$t_r \leq t_r^{ref}$$

- $\texttt{MR}_{osc}$: If pressure oscillations are present, the number of speed oscillations should be below or equal to that of the reference.

$$\text{If } \texttt{osc} \leq osc^{ref}$$

- $\texttt{MR}_{ess_g}$: The ESS should be greater than or equal to that of the reference.

$$ESS \geq ESS_{ref}$$

- $\texttt{MR}_{ess_l}$: The steady-state error should be less than or equal to that of the reference.

$$ESS < ESS_{ref}$$

- $\texttt{MR}_{IAE_l}$: The IAE should be smaller than or equal to that of the reference.

$$IAE \leq IAE_{ref}$$

- $\texttt{MR}_{IAE_g}$: The IAE should be greater than or equal to that of the reference (expected under degraded performance).

$$IAE \geq IAE_{ref}$$

- $\texttt{MR}_{ITAE_l}$: The Integral of ITAE should be smaller than or equal to that of the reference.

$$ITAE \leq ITAE_{ref}$$

- $\texttt{MR}_{ITAE_g}$: The ITAE should be greater than or equal to that of the reference.

$$ITAE > ITAE_{ref}$$

- $\texttt{MR}_{Eng_{ratio-g}}$: The ratio of pressure energy to speed energy should increase or remain constant.

$$\frac{Eng_p}{Eng_s} \geq \left(\frac{Eng_p}{Eng_s}\right)_{ref}$$

Table 5: Summary of MRs for selected PID parameter modification patterns. Each row shows a specific combination of K_p, K_i, and K_d parameter changes—denoted as `L` (lower), `E` (equal), and `H` (higher)—and the corresponding set of MRs used to verify changes in output behaviour.

Change in the input	**Expected output change**	**Total MRs**
`LLL`	$\texttt{MR}_{RT}$, $\texttt{MR}_{PR}$, $\texttt{MR}_{Eng_{ratio-l}}$, $\texttt{MR}_{osc}$, $\texttt{MR}_{IAE}$	5
`HLH`	$\texttt{MR}_{RT}$, $\texttt{MR}_{PR}$, $\texttt{MR}_{ess_g}$, $\texttt{MR}_{Eng_{ratio-g}}$	4
`HLL`	$\texttt{MR}_{RT}$, $\texttt{MR}_{PR}$, $\texttt{MR}_{ess_g}$, $\texttt{MR}_{IAE_g}$, $\texttt{MR}_{ITAE_g}$, $\texttt{MR}_{Eng_{ratio-g}}$	6
`LHL, EHL`	$\texttt{MR}_{RT}$, $\texttt{MR}_{PR}$, $\texttt{MR}_{ess_l}$, $\texttt{MR}_{IAE_g}$, $\texttt{MR}_{ITAE_g}$, $\texttt{MR}_{Eng_{ratio-l}}$	6
`ELL, LLH, EEL, ELH`	$\texttt{MR}_{RT}$, $\texttt{MR}_{PR}$, $\texttt{MR}_{ess_g}$, $\texttt{MR}_{IAE_g}$, $\texttt{MR}_{ITAE_g}$, $\texttt{MR}_{Eng_{ratio-l}}$	6
`HHL`	$\texttt{MR}_{RT}$, $\texttt{MR}_{PR}$, $\texttt{MR}_{ess_l}$, $\texttt{MR}_{ITAE_g}$, $\texttt{MR}_{Eng_{ratio-g}}$	5
`HHH`	$\texttt{MR}_{RT}$, $\texttt{MR}_{PR}$, $\texttt{MR}_{ess_l}$, $\texttt{MR}_{IAE_l}$, $\texttt{MR}_{ITAE_l}$, $\texttt{MR}_{Eng_{ratio-l}}$	6
`EHH, HEH`	$\texttt{MR}_{RT}$, $\texttt{MR}_{PR}$, $\texttt{MR}_{ess_l}$, $\texttt{MR}_{IAE_l}$, $\texttt{MR}_{ITAE_l}$, $\texttt{MR}_{Eng_{ratio-g}}$	6

- $\texttt{MR}_{Eng_{ratio-l}}$: The ratio of pressure to speed energy should decrease or remain constant.

$$\frac{Eng_p}{Eng_s} < \left(\frac{Eng_p}{Eng_s}\right)_{ref}$$

We developed a Python script that encodes the MRs as conditional rules. Each MR is implemented as a function that evaluates whether a given parameter combination satisfies the expected behavioural trend by returning 1, or does not satisfy it by returning 0. The script processes the simulation output data and automatically checks each parameter set against the defined MRs. Additionally, it generates detailed logs and diagnostic plots to support further analysis. The interpretation of the MR evaluation results and their implications for parameter selection are discussed in detail in Section 6.

6. Discussion

In this discussion, we focus exclusively on parameter configurations that satisfy the $\texttt{MR}_{PR}$, which serves as a necessary precondition for meaningful analysis. $\texttt{MR}_{PR}$ validates that the system's output behaviour remains within the allowed OS threshold in relation to the target. Since our objective is to identify and reason about optimal or near-optimal response profiles, only samples that satisfy $\texttt{MR}_{PR}$ are included in the

analysis. This acts as an initial filter that ensures all examined parameter configurations demonstrate at least a baseline level of acceptable behaviour. Consequently, all results reported in the remainder of the discussion refer to samples with at least one satisfied MR—specifically, $\texttt{MR}_{PR}$.

6.1. Key Findings

Figure 4 shows the distribution of MR satisfaction across different gain patterns for each machine configuration. In $\texttt{tc}_{100}$, gain patterns such as LLL, LLH, and ELL demonstrate high median MR satisfaction, with several parameter configurations satisfying all 5 available MRs. In contrast, gain patterns like EHH, EEL, and HEH show low MR satisfaction, often with zero or near-zero values.

For $\texttt{tc}_{104}$, we observe a wider spread. Patterns like ELH, EHH, and HHH show stronger MR satisfaction (often reaching 5 or 6 MRs), reinforcing their potential as robust configurations under changing system dynamics. In $\texttt{tc}_{106}$, the spread tightens considerably. Most gain patterns have very low MR satisfaction, with median values near or below 2. Exceptions include LLL and ELL, which still reach up to 4 MRs, standing out as resilient under this more challenging configuration. Gain patterns with predominantly low parameter values (LLL, LLH, ELL) tend to consistently satisfy more MRs, regardless of the machine configuration. Performance is scenario-dependent: some gain patterns perform well in $\texttt{tc}_{100}$ and $\texttt{tc}_{104}$ but degrade under $\texttt{tc}_{106}$, highlighting the importance of analysing configurations across diverse test inputs.

Table 6 complements the MR satisfaction distributions by highlighting only those parameter configurations that achieved the maximum number of satisfied MRs for each scenario. For instance, in configuration LLL under $\texttt{tc}_{100}$, 21 parameter configurations satisfied all 5 available MRs, suggesting that this combination yields 21 potentially optimal parameter sets. The MRs column indicates this maximum (*e.g.,* 5 or 6), while the Count reflects how many parameter configurations reached that peak.

To better understand why these parameter configurations are promising, Figure 5 presents the system responses (pressure and speed) over time for randomly selected LLL samples across the three machine configurations. Each subplot shows the pressure (top) and speed (bottom) response in comparison to the corresponding target signal (in red). Despite variations in scenario complexity, the majority of LLL parameter sets produce smooth, stable responses with minimal overshooting and fast settling. These qualitative observations provide further evidence that high MR satisfaction is a reliable indicator of desirable dynamic behaviour and reinforce the role of the LLL configuration as a strong candidate for optimal control.

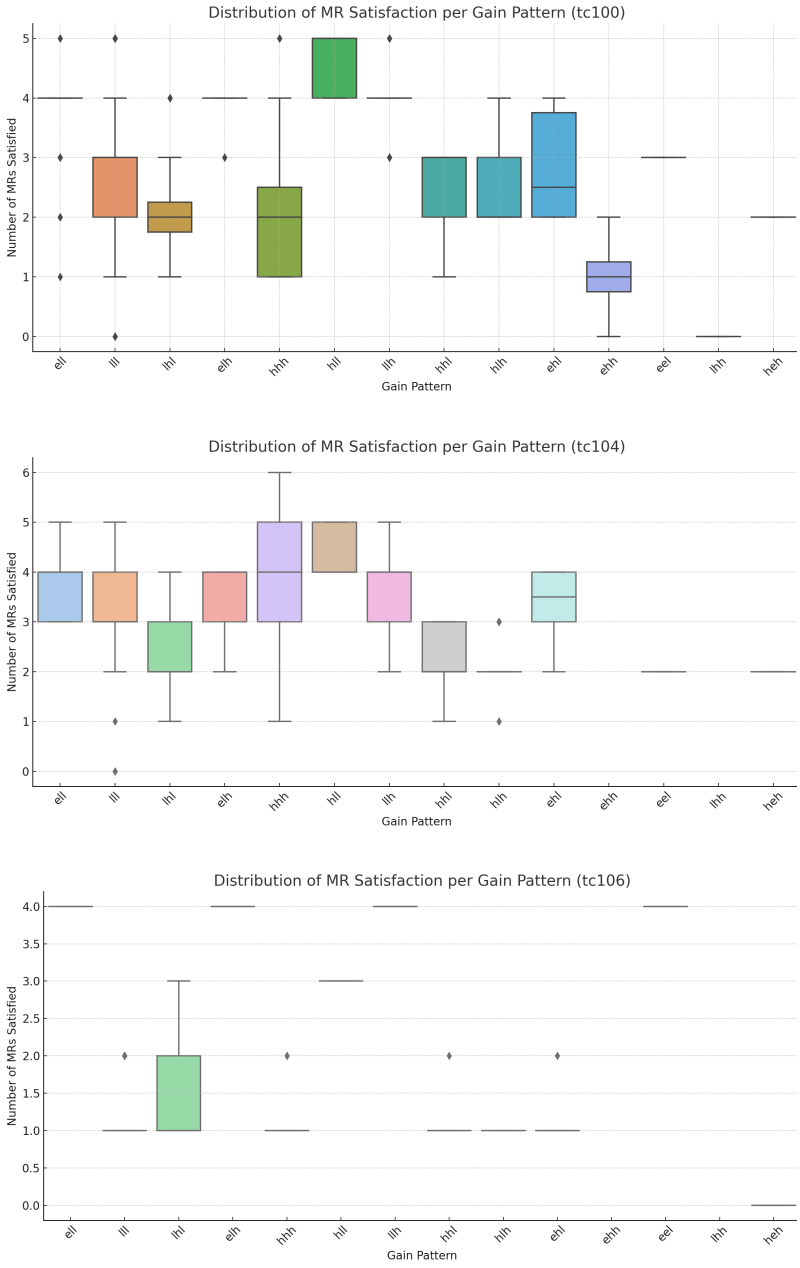

Figure 4: Distribution of the number of satisfied MRs for each gain pattern across the three machine configurations: $\texttt{tc}_{100}$, $\texttt{tc}_{104}$, and $\texttt{tc}_{106}$. Each boxplot summarises the spread and central tendency of MR satisfaction among parameter configurations for a given PID parameter combination.

Table 6: Summary of gain patterns producing parameter configurations with maximum MR satisfaction. The MRs column indicates the maximum number of MRs satisfied by any test case under the respective combination, also which the information on how many parameter configurations were observed to satisfying the MRs in the column #.

Mostly decreased								**Mostly increased**								**Mixed changes**							
		tc_{100}		tc_{104}		tc_{106}				tc_{100}		tc_{104}		tc_{106}				tc_{100}		tc_{104}		tc_{106}	
Com	**#**	**MRs**	**#**	**MRs**	**#**	**MRs**	**#**	**Com**	**#**	**MRs**	**#**	**MRs**	**#**	**MRs**	**#**	**Com**	**#**	**MRs**	**#**	**MRs**	**#**	**MRs**	**#**
LLL	601	5	21	5	36	2	12	HHH	23	5	3	6	3	2	2	ELH	21	4	20	3	1	-	-
HLL	86	-	-	-	-	-	-	HLH	92	-	-	2	1	2	1	EHL	6	-	-	4	4	2	1
LHL	52	4	3	4	8	3	1	HHL	29	3	-	-	-	-	-								
ELL	45	5	2	-	-	-	-	EHH	4	-	-	4	2	1	1								
LLH	33	5	5	5	1	-	-	LHH	7	-	-	-	-	-	-								
EEL	1	3	1	-	-	-	-	HEH	1	2	1	2	1	-	-								

For the sake of reproducibility, the scripts used to automatically generate these plots—as well as the complete set of response plots used in this analysis—are available in our project repository. Additionally, we are in the process of developing a lightweight tool designed to support the fast visualization and analysis of PID behaviours. This tool, which can be used to reproduce and explore the results presented in this paper, is available in our GitHub repo[3].

During our manual exploration of the gain pattern groups, we observed that parameter configurations satisfying a similar number of MRs often exhibit qualitatively similar system responses. This internal consistency was particularly evident within the same gain pattern or satisfaction level group (*e.g.,* all parameter configurations satisfying 4 MRs in `LLL`). However, these patterns were identified purely through visual inspection, without an automated clustering process. Interestingly, we also observed frequent inconsistencies between quantitative metrics such as energy consumption and the sum of ESS. In many cases, a response with lower energy did not necessarily yield a lower ESS, and vice versa. This inconsistency likely stems from the fundamentally different aspects each metric captures.

While ESS quantifies how closely a response follows the target trajectory over time, energy-based metrics penalise the intensity or effort of the response. A system

[3]`https://github.com/aduquet/Tool-visualizingSimulationRuns`

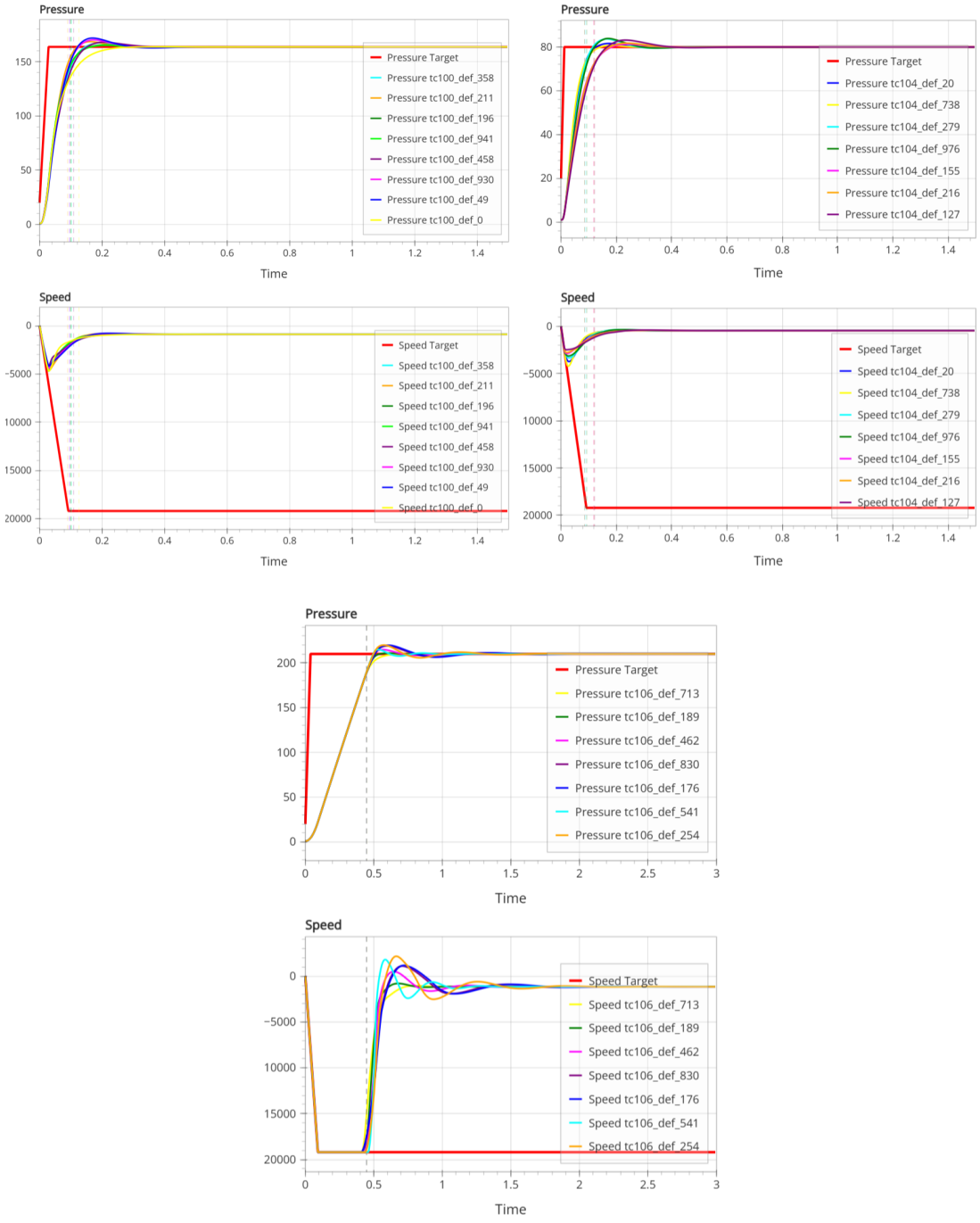

Figure 5: Behaviour of random samples from the LLL gain pattern group across three machine configurations (tc100, tc104, tc106).

may therefore exhibit a smooth, energy-efficient behaviour while deviating from the target (yielding a high ESS), or track the target tightly but with excessive control effort (resulting in high energy). As a result, we observed several cases where the MR related to energy was satisfied while the MR related to ESS was not, and vice versa. There were also cases where both MRs were satisfied simultaneously.

To validate our approach, we asked our industry partner to select the top 10 parameter configurations they considered most desirable, based on their domain expertise and visual inspection. Their selection, made for the `tc100` scenario, included test IDs 471, 801, and 22 (all from the `HHH` group), as well as 757, 882, 361, 932, 277, 271, and 551 (all from the `LLL` group). As shown in Figure 6, these parameter configurations display desirable dynamic behaviours, including fast rise times, low overshoot, and smooth convergence to the target in both pressure and speed. Interestingly, the `HHH` samples satisfied 5 out of 6 MRs—placing them in the top MR satisfaction group—while the selected `LLL` cases, though preferred by the partner, belonged to the next tier, satisfying 4 out of 5 MRs. Further inspection revealed that this gap often resulted from trade-offs between the energy and IAE metrics: some parameter configurations satisfied the energy-related MR but failed the IAE-related one, and vice versa. These examples illustrate how expert judgment tends to align with certain MRs (*e.g.,* MR_{PR}, MR_{osc}), but may diverge from others when underlying performance trade-offs are involved—highlighting the need for balanced evaluation criteria.

Energy-based MRs are designed to assess whether a response maintains smooth dynamics while avoiding excessive oscillations or unnecessarily high control effort. In contrast, IAE-related MRs focus on the cumulative deviation from the target over time, which means fast error correction. As a result, a controller that limits energy usage may exhibit slower or more conservative behaviour, which can lead to higher accumulated error and failure to satisfy the IAE criterion. Conversely, a controller that minimises IAE may achieve tighter tracking at the expense of increased control activity, potentially violating the energy-related MRs. This trade-off between different performance aspects helps explain why several expert-preferred samples, although visually optimal, did not satisfy all MRs simultaneously.

Our analysis confirms that MR satisfaction—particularly when evaluated across diverse machine configurations—offers a meaningful proxy for identifying potentially optimal parameter configurations. Patterns such as `LLL` consistently demonstrated strong performance both quantitatively (in terms of MR satisfaction) and qualitatively (in their dynamic behaviour). However, inconsistencies between metrics like energy and IAE and ESS revealed trade-offs that individual MRs alone may not fully capture. Expert-selected samples further highlighted this tension, underscoring

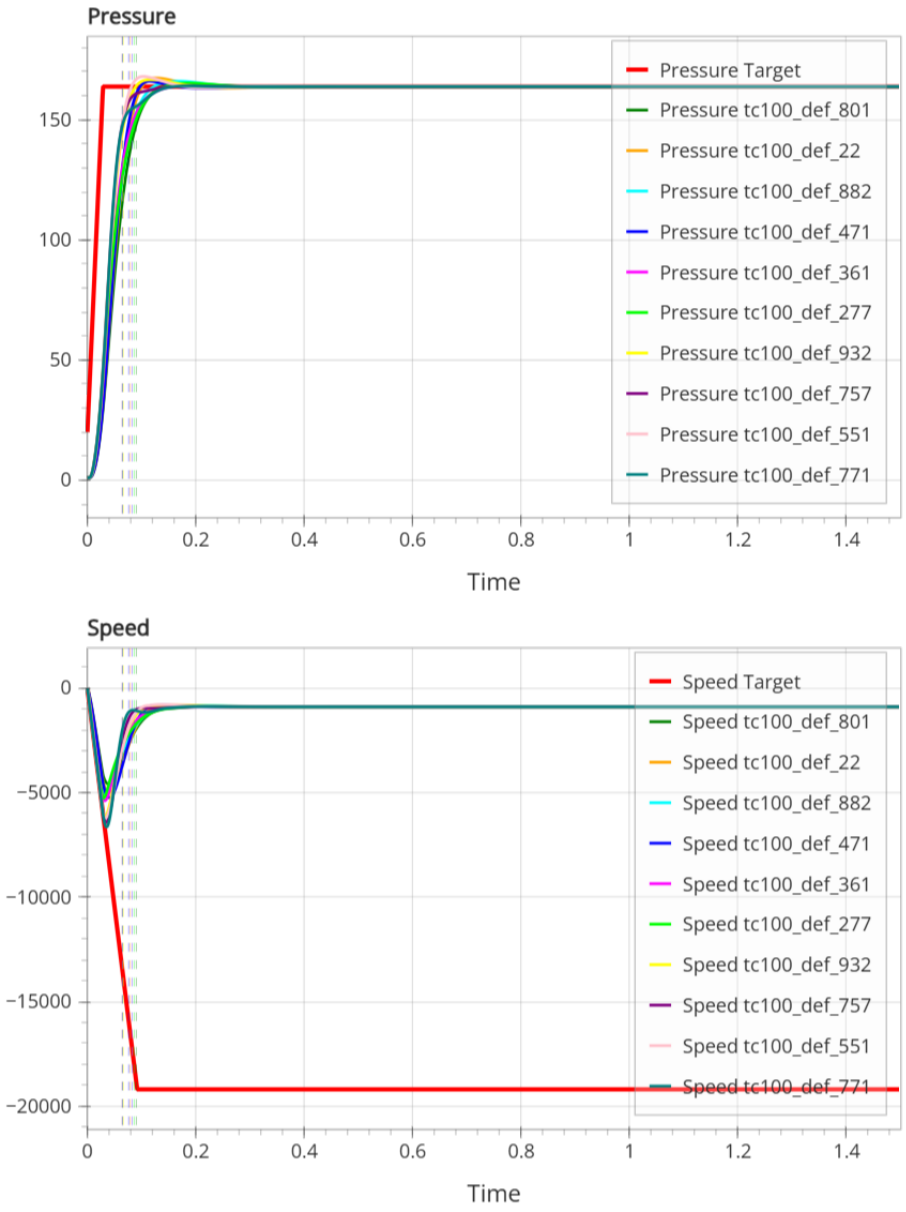

Figure 6: Pressure (top) and speed (bottom) response of the top 10 parameter configurations selected by the industry partner for $\texttt{tc}_{100}$. These include samples from both the `HHH` and `LLL` gain patterns. All responses are plotted against the respective target signal (in red). Most selected parameter configurations exhibit smooth and stable behaviours, closely following the target trajectory.

the importance of considering multiple perspectives when evaluating control quality. These findings point to the potential of MR-based evaluation as a practical guide for controller design, while also highlighting the need for refined or composite MRs that better reconcile conflicting performance goals.

6.2. Future Directions

This study opens the pathway for deeper investigation into MT for control systems. One promising direction is the integration of runtime monitoring information to enhance the MRs. Rather than treating MR satisfaction as a binary outcome, future work could explore graded or softness MR evaluation, where degrees of compliance are tracked over time or within defined tolerances. This refinement would support the detection of near-miss faults—subtle behaviours that deviate slightly from expected trends but may still impact performance. One open question is how to balance the flexibility of more tolerant MRs with the precision of stricter ones. Allowing limited deviations might help identify near-optimal behaviours that strict MRs would reject, but also increases the risk of accepting suboptimal responses. Future research should examine how varying levels of MR strictness affect the reliability of the testing process and whether introducing thresholds improves or weakens fault detection capability.

One limitation we identified is that the interpretation of MR results currently involves significant manual effort. To address this, we plan to combine the MR-based evaluation with an automated ranking system, capable of prioritising parameter configurations based on metric interactions, MR satisfaction patterns, and expert-informed preferences. This integration aims to improve the interpretability and practical usability of the approach, supporting more efficient decision-making in real-world testing and tuning processes.

6.3. Threats to Validity

In our study, four types of threats to validity are most relevant: threats to construct, internal, external, and reliability validity.

Construct validity concerns the appropriateness of our definitions and the way we interpret MR satisfaction as a proxy for system performance. The assumption that higher MR satisfaction corresponds to more desirable dynamic behaviour may not always hold, particularly when MRs focus on different and potentially conflicting metrics (*e.g.,* energy vs. IAE). We mitigated this threat by including qualitative visual inspections and incorporating expert-selected parameter configurations as a secondary validation layer. Another threat is the manual selection and definition

of MRs, which inherently involves subjectivity. To address this, our MRs were derived from both domain-specific control knowledge (*e.g.*, acceptable OS, rising time behaviour) and general control performance indicators.

Internal validity relates to the experimental setup, including the quality and diversity of the test data used. Our PID parameter configurations were generated using a combinatorial sampling strategy in combination with the Student's t-distribution centred around a baseline configuration, ensuring a broad yet relevant coverage of the parameter space. However, some gain patterns (*e.g.,* those with extreme values) may be under-represented, which could influence observed MR satisfaction trends.

External validity refers to the generalisability of our findings beyond the studied machine configurations and system. Our approach was applied to three machine configurations derived from a single closed-loop PID-controlled system. Although these scenarios vary in complexity and disturbance levels, all results remain tied to a specific plant model and simulation setup. The generalisability of the proposed method to other systems (*e.g.,* multi-variable controllers, non-linear plants) remains to be confirmed. That said, the methodology—combining MR satisfaction analysis with runtime behaviour interpretation—can be adapted to other control systems and evaluation contexts.

Reliability validity concerns the reproducibility of our results and the transparency of the evaluation process. While the core analysis was conducted using scripts and tooling developed for this study, some components—such as visual inspection or grouping by behavioural trends—still require manual effort. To address this, we provide the scripts used for generating MR satisfaction plots, the full set of response plots, and a tool under development for visualising and exploring simulation runs. These resources improve reproducibility and offer transparency into our evaluation pipeline.

7. Conclusion

This paper presents our experience in using MT for supporting the search for improved PID controller parameter configurations. Rather than relying solely on expert heuristics or optimisation algorithms, our approach uses domain-informed MRs to systematically assess the dynamic behaviour of the system under varying gain combinations. By analysing which configurations satisfy predefined MRs, we are able to identify candidate parameter sets that demonstrate desirable control properties such as stability, low overshoot, fast settling, and efficient energy usage.

Our results indicate that MR satisfaction provides a reliable indicator for distinguishing promising parameter configurations. For example, gain patterns like

LLL consistently achieved high MR satisfaction across scenarios, and many expert-preferred samples also aligned with MR-driven assessments. While it is true that none of the expert-selected configurations satisfied all MRs, it is important to note that several—such as the LLL configurations in the $\texttt{tc}_{100}$ scenario—still satisfied 4 out of 5 MRs, placing them in the upper tier of MR satisfaction. This suggests that the MRs are effectively capturing relevant and desirable behavioural trends, even if they do not perfectly mirror expert preferences in every case. This observation highlights its practical value as a pre-selection mechanism: configurations that satisfy a high number of MRs are likely to exhibit favourable behaviours and merit closer inspection. However, the fact that some high-MR configurations were not selected, and vice versa, points to the existence of context-specific trade-offs—for example, between energy efficiency and precision—that are better evaluated through human expertise.

We also identified important limitations. Interpreting MR results remains a manual and time-consuming task, which poses challenges for scalability in larger systems. Additionally, trade-offs between certain performance metrics—such as energy consumption and error correction—highlight the need for evaluation strategies that can effectively handle competing objectives.

To overcome these issues, future work will focus on (i) incorporating runtime monitoring data into the definition and application of MRs, (ii) investigating graded MR satisfaction to better capture borderline or near-optimal behaviours, and (iii) integrating MR analysis with automated ranking tools to streamline the decision-making process.

8. Acknowledgements

The research reported in this article has been partly funded by Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK), the Federal Ministry for Labour and Economy (BMAW), and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE [(FFG grant no. 892418)] in the COMET - Competence Centers for Excellent Technologies Programme managed by Austrian Research Promotion Agency FFG, as well as by the European Regional Development Fund, and grant PRG1226 of the Estonian Research Council.

References

[1] A. Riccardi, E. Minisci, K. Akartunali, C. Greco, N. Rutledge, A. Kershaw, A. Hashim, Introduction to optimisation, in: Optimization Under Uncer-

tainty with Applications to Aerospace Engineering, Springer, 2020, pp. 223–268. URL: https://link.springer.com/chapter/10.1007/978-3-030-60166-9_7. doi:10.1007/978-3-030-60166-9_7.

[2] M. Y. Coskun, M. İtik, Intelligent pid control of an industrial electro-hydraulic system, ISA Transactions 139 (2023) 484–498. URL: https://www.sciencedirect.com/science/article/pii/S0019057823001751. doi:https://doi.org/10.1016/j.isatra.2023.04.005.

[3] T. L. Blevins, Pid advances in industrial control, IFAC Proceedings Volumes 45 (2012) 23–28.

[4] R. Benotsmane, G. Kovács, Optimization of energy consumption of industrial robots using classical pid and mpc controllers, Energies (2023). URL: https://www.mdpi.com/1996-1073/16/8/3499. doi:10.3390/en16083499.

[5] Y. A. Ali, E. M. Awwad, M. Al-Razgan, A. Maarouf, Hyperparameter search for machine learning algorithms for optimizing the computational complexity, Processes 11 (2023). URL: https://www.mdpi.com/2227-9717/11/2/349. doi:10.3390/pr11020349.

[6] A. Morales-Hernández, I. Van Nieuwenhuyse, S. Rojas Gonzalez, A survey on multi-objective hyperparameter optimization algorithms for machine learning, Artificial Intelligence Review 56 (2023) 8043–8093. URL: https://link.springer.com/article/10.1007/s10462-022-10359-2. doi:10.1007/s10462-022-10359-2.

[7] S. Formentin, K. van Heusden, A. Karimi, A comparison of model-based and data-driven controller tuning, International Journal of Adaptive Control and Signal Processing 28 (2014) 882–897.

[8] F. Lin, R. Brandt, G. Saikalis, Self-tuning of pid controllers by adaptive interaction, in: Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334), volume 5, 2000, pp. 3676–3681 vol.5. doi:10.1109/ACC.2000.879256.

[9] O. T. Altinoz, H. Erdem, Particle swarm optimisation-based pid controller tuning for static power converters, International Journal of Power Electronics 7 (2015) 16–35.

[10] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

[11] F. Hutter, H. H. Hoos, K. Leyton-Brown, Sequential model-based optimization for general algorithm configuration, in: C. A. C. Coello (Ed.), Learning and Intelligent Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 507–523.

[12] J. B. Mockus, L. J. Mockus, Bayesian approach to global optimization and application to multiobjective and constrained problems, J. Optim. Theory Appl. 70 (1991) 157–172.

[13] Sensitivity Analysis in Diagnostic Modelling: Monte Carlo Filtering and Regionalised Sensitivity Analysis, Bayesian Uncertainty Estimation and Global Sensitivity Analysis, John Wiley and Sons, Ltd, 2002, pp. 151–192.

[14] J. Kapinski, J. V. Deshmukh, X. Jin, H. Ito, K. Butts, Simulation-based approaches for verification of embedded control systems: An overview of traditional and advanced modeling, testing, and verification techniques, IEEE Control Systems Magazine 36 (2016) 45–64. doi:10.1109/MCS.2016.2602089.

[15] W. Ho, H. Wong, H. Han, L. Ni, Robustness monitoring for pid control systems, in: IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200), volume 3, 1998, pp. 1422–1426 vol.3. doi:10.1109/IECON.1998.722860.

[16] X. Li, C. Li, Y. Hu, Y. Yu, W. Zeng, H. Wu, Uncertainty quantification of the power control system of a small pwr with coolant temperature perturbation, Nuclear Engineering and Technology 54 (2022) 2048–2054.

[17] T. Y. Chen, S. C. Cheung, S. M. Yiu, Metamorphic testing: a new approach for generating next test cases, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, Tech. Rep. (1998).

[18] A. Duque-Torres, D. Pfahl, C. Klammer, S. Fischer, Using source code metrics for predicting metamorphic relations at method level, in: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE Computer Society, Los Alamitos, CA, USA, 2022, pp. 1147–1154.

[19] A. Duque-Torres, D. Pfahl, C. Klammer, S. Fischer, Bug or not bug? analysing the reasons behind metamorphic relation violations, in: IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2023, pp. 905–912.

[20] S. Singcuna, P. Kittisupakorn, D. Banjerdpongchai, J. Dawkrajai, P. Daskalov, T. Georgieva, Implementing pid control on plc benchmark to enhance industrial automation skills towards asean factori 4.0, in: 2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), 2023, pp. 409–414. doi:10.1109/ECTIDAMTNCON57770.2023.10139688.

[21] C. Knospe, Pid control, IEEE Control Systems Magazine (2006) 30–31. doi:10.1109/MCS.2006.1580151.

[22] R. P. Borase, D. K. Maghade, S. Y. Sondkar, S. N. Pawar, A review of pid control, tuning methods and applications, International Journal of Dynamics and Control 9 (2021) 818–827. URL: https://doi.org/10.1007/s40435-020-00665-4. doi:10.1007/s40435-020-00665-4.

[23] P. M. Meshram, R. G. Kanojiya, Tuning of pid controller using ziegler-nichols method for speed control of dc motor, in: IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012), 2012, pp. 117–122.

[24] A. Utami, R. Yuniar, A. Giyantara, A. Saputra, Cohen-coon pid tuning method for self-balancing robot, in: 2022 International Symposium on Electronics and Smart Devices (ISESD), 2022, pp. 1–5. doi:10.1109/ISESD56103.2022.9980830.

[25] F. Isdaryani, F. Feriyonika, R. Ferdiansyah, Comparison of ziegler-nichols and cohen coon tuning method for magnetic levitation control system, Journal of Physics: Conference Series 1450 (2020) 012033. URL: https://dx.doi.org/10.1088/1742-6596/1450/1/012033. doi:10.1088/1742-6596/1450/1/012033.

[26] M. Campi, S. Savaresi, Direct nonlinear control design: the virtual reference feedback tuning (vrft) approach, IEEE Transactions on Automatic Control 51 (2006) 14–27. doi:10.1109/TAC.2005.861689.

[27] S. Soma, O. Kaneko, T. Fujii, A new method of controller parameter tuning based on input-output data – fictitious reference iterative tuning (frit), IFAC Proceedings Volumes 37 (2004) 789–794. IFAC Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP 04) and IFAC Workshop

on Periodic Control Systems (PSYCO 04), Yokohama, Japan, 30 August - 1 September, 2004.

[28] P. Gil, A. Sebastião, C. Lucena, Constrained nonlinear-based optimisation applied to fuzzy pid controllers tuning, Asian Journal of Control 20 (2018) 135–148. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/asjc.1549. doi:https://doi.org/10.1002/asjc.1549. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asjc.1549.

[29] X. Zuo, J.-w. Liu, X. Wang, H.-q. Liang, Adaptive pid and model reference adaptive control switch controller for nonlinear hydraulic actuator, Mathematical Problems in Engineering 2017 (2017).

[30] J. L. Meza, V. Santibanez, R. Soto, M. A. Llama, Fuzzy self-tuning pid semiglobal regulator for robot manipulators, IEEE Transactions on Industrial Electronics 59 (2012) 2709–2717. doi:10.1109/TIE.2011.2168789.

[31] S. Documentation, Simulation and model-based design, 2020. URL: https://www.mathworks.com/products/simulink.html.

[32] S. AG, Totally Integrated Automation Portal (TIA Portal), 2025. URL: https://www.siemens.com/us/en/products/automation/industry-software/automation-software/tia-portal.html, accessed: 2025-03-10.

[33] R. Automation, Studio 5000 Design Software, 2025. URL: https://www.rockwellautomation.com/en-us/products/software/factorytalk/designsuite/studio-5000.html, accessed: 2025-03-10.

[34] S. Segura, G. Fraser, A. B. Sánchez, A. Ruiz-Cortés, A survey on metamorphic testing, IEEE Transactions on Software Engineering 42 (2016) 805–824. doi:10.1109/TSE.2016.2532875.

[35] N. Kalra, N. Moehle, S. A. Seshia, Testing cyber-physical systems using metamorphic relations, in: Proceedings of the 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS), 2020, pp. 187–198. doi:10.1109/ICCPS48487.2020.00022.

[36] Z. Cao, M. Jeon, Y. Xu, K. Leach, W. Weimer, Metamorphic testing for quality assurance of neural control systems, in: Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2021, pp. 912–924. doi:10.1109/ASE51524.2021.9678859.

[37] T. Y. Chen, F.-C. Kuo, W. K. Tam, R. G. Merkel, Testing a software-based pid controller using metamorphic testing, in: PECCS, 2011, pp. 387–396.

[38] F.-C. Kuo, T. Y. Chen, W.-K. Tam, Testing proportional-integral-derivative (pid) controller with metamorphic testing, International Journal of Software Engineering and Knowledge Engineering 22 (2012) 1143–1165.