Application of Machine Learning to Performance Assessment for a class of PID-based Control Systems

Patryk Grelewicz, Thanh Tung Khuat, Jacek Czeczot, Tomasz Klopot, Bogdan Gabrys

***Abstract*—In this paper, a novel machine learning derived control performance assesment (CPA) classification system is proposed. It is dedicated for a class of PID-based control loops with processes exhibiting second order plus delay time (SOPDT) dynamical properties. The proposed concept is based on deriving and combining a number of different, diverse control performance indices (CPIs) that separately do not provide sufficient information about the control performance. However, when combined together and used as discriminative features of the assessed control system, they can provide consistent and accurate CPA information. This concept is discussed in terms of the introduced extended set of CPIs, comprehensive performance assessment of different machine learning based classification methods and practical applicability of the suggested solution. The latter is shown and verified by practical application of the proposed approach to a CPA system for a laboratory heat exchange and ditribution setup.**

***Index Terms*—Control Performance Assessment, Practical implementation, Programmable logic controllers, PID control, Cloud computing, Machine learning, Pattern Classification, Diagnostic Analysis.**

I. Introduction

IN modern industrial control systems, high control performance of low-level controllers is crucial for efficient process operation [1]. This high performance is usually ensured by proper design [2]-[3] and tuning [4] of the controllers, e.g. using virtual commissioning approaches [5]-[6]. However, the quality of the control usually degrades over time due to fluctuations of process dynamics resulting e.g. from slow fouling, slow decrease in accuracy of sensors and actuators, modifications in production conditions, etc. [7]. Consequently, performance of over 60% of control loops is poor [8] and in vast majority of cases it results from a bad tuning of the controllers [9]. Thus, periodical control performance assessment becomes more and more important and is necessary to meet the requirements of Industry 4.0 in terms of preserving the best process efficiency [10]-[11].

Many control performance assessment (CPA) algorithms have been developed over last decades and they gained popularity among academia [12]-[13] and industry [14]-[15]. The first group is based on performing a comparison between the current control performance and the best observed so far in terms of variance of manipulating and process variables [16]-[19]. These methods are based on normalized indices and their interpretation is clear. However, there is no explicit classification if control performance is acceptable or not and how much this performance can be improved. Additionally, results depend strongly on stochastic characteristics of process disturbances. Thus, these CPA algorithms can be used for monitoring degradation in control performance but not for its absolute assessment. The second group is based on deriving and using general control performance indices (CPIs) that can be calculated for certain deterministic properties of a control system like a set point tracking and/or disturbance rejection. Based on time responses, different CPIs can be proposed, such as e.g. settling time, maximum overshoot, absolute square error, etc. [7] and it has already been shown that there exists a correlation between their values and variance-based performance measures [20]. Application of these CPIs has been suggested for quantitative comparison between different controllers and/or different tunings but still, there is a lack of general rules how to use them for an explicit CPA.

In this paper, this research gap is tackled by proposing a machine learning derived CPA classification system in the application to conventional PID-based control loops working on a broad class of processes exhibiting second order+delay time (SOPDT) dynamical properties. In industrial practice, the PID controller is still the most frequently used in low-level control loops and its application to control processes accurately approximated by SOPDT dynamics is very common. The proposed CPA system is based on leveraging the predefined benchmark disturbance rejection response of control system subject to SOPDT parameters and optimal PID tuning. The acceptable deviation of this response is defined and training dataset is generated by systematically simulating

This work was financed in part by the grant from SUT - subsidy for maintaining and developing the research potential in 2020 (J. Czeczot, T. Klopot), and by BKM grant (BKM-723/RAU3/2020) and co-financed by the European Union through the European Social Fund (grant POWR.03.02.00-00-I029) (P. Grelewicz). Calculations were done with the use of GeCONiI infrastructure (POIG 02.03.01-24-099).

P. Grelewicz, J. Czeczot and T. Klopot are with Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, Department of Automatic Control and Robotics, Gliwice, Poland.

TT. Khuat and B. Gabrys are with University of Technology Sydney, Faculty of Engineering and IT, Advanced Analytics Institute, New South Wales 2007, Australia.

J. Czeczot is the corresponding author (jacek.czeczot@polsl.pl).

and recording acceptable and not acceptable disturbance rejection responses together with a set of related CPIs calculated from these responses. Once generated, this training dataset is used to train machine learning based classifiers to find accurate mapping between the CPIs and the class label (i.e. if the quality of control is acceptable or not). As part of the analysis of the feasibility and accuracy of such a mapping and its usefulness in control settings, a comprehensive comparative analysis of a wide range of ML based classification algorithms and an assessment of useful discriminative information contained in the proposed set of CPIs are also performed.. Finally, practical cloud-based implementation of this system for PLC-based control loop is presented and experimental results show practical applicability of the suggested concept.

The major novelty of this paper results from the following contributions:

- introducing the concept of machine learning-based CPA system for PID+SOPDT control loops,
- proposing the method for deriving a training dataset to ensure successful training of selected classifiers,
- defining and proposing of a substantially extended set of CPIs used to accurately capture the nuances of the control system response to th step change of load disturnace,
- comprehensive, comparative analysis of different machine learning algorithms performance and their applicability for the proposed CPA system,
- practical implementation of the suggested CPA system and its validation in the application to PID-based control system implemented in PLC and applied to control laboratory heat exchange and distribution setup.

II. Statement of the Problem

This study concentrates on the CPA of closed-loop control systems shown in Fig. 1 with the conventional PID controller and stable time-invariant process exhibiting (SOPDT) dynamics. The control goal is defined to keep the process output *y* at a set point *sp* in the presence of an additive load disturbance *d*. In process automation, vast majority of control systems are designed to provide an affective disturbance rejection so the CPA is limited to this case. For this purpose, it is assumed that it is acceptable to apply a small step change Δd to excite the process input and to initialize collecting data for the CPA.

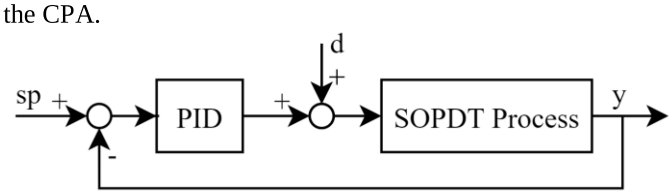

Fig. 1 Considered PID-based closed loop system with load disturbance *d*.

A normalization of different SOPDT dynamics is made by introducing two normalized dynamical parameters $L_1 = \tau_0/(\tau_1 + \tau_0)$ and $L_2 = \tau_2/\tau_1$ where $\tau_1 \geq \tau_2$ and τ_0 denote two process time constants and a delay time, respectively. This paper concentrates on the CPA for SOPDT processes with unitary (normalized) gain, for which $L_1 \in [0.1, 0.6]$ and $L_2 \in [0.1, 1.0]$. Note that these processes can be efficiently controlled by a conventional PID controller. For $L_1>0.6$, the delay time is dominant and more advanced control strategies are suggested. At the same time, for L_1, $L_2 < 0.1$, a conventional PI controller can be easily tuned and applied.

For the considered control system, it is assumed that the CPA is based only on the values of selected CPIs that are computed from the response to the applied disturbance step change Δd. Its amplitude should be adjusted to ensure significant process excitation but to prevent from inadmissible process disturbing. In general, CPIs are relatively easy for on-line computation but their selective use for the CPA is very limited. Their values depend strongly on process dynamics but this problem can be effectively managed by appropriate scaling [21]. Much more important is the fact that they *individually* focus only on very limited properties of the dynamical response, which is illustrated in Fig. 2 for three differently tuned examples of PID controllers within the same control system (denoted as CS1, CS2 and CS3). Note that CS2 outperforms CS1 in terms of the overshoot but CPIs that focus on oscillatory behaviour and settling time are clearly better for CS1. Only a fusion of different CPIs can give more reliable information on control performance. However, even then, CPA based on CPIs is still a challenging task. Time responses for CS1 and CS3 do not give a clear indication which of the two controllers performe better. CS1 has the worst overshoot but its settling time is comparable to CS3 without any oscillations. Thus, the final choice should be made based on technological requirements or predefined properties of control system assumed as optimal for the considered SOPDT process determined by L_1, L_2. In this paper, the latter approach is considered.

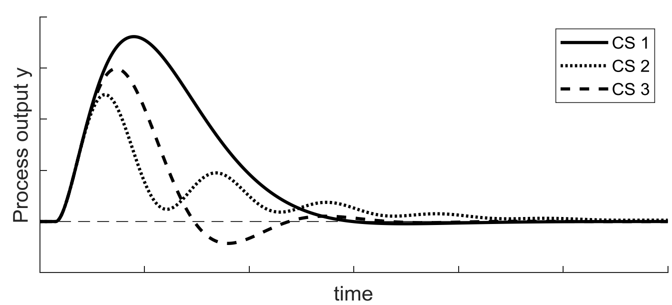

Fig. 2. Illustrative examples of responses of three differently tuned control systems to a step change of the load disturbance.

III. CPA for PID+SOPDT Control Systems

The CPA problem defined in section II is proposed to be tackled and solved by designing a binary classifier based on a supervised machine learning approach. This concept is based on the thesis that a sufficiently large number of different CPIs capturing diverse aspects of the time series representing the control system's response to a step change of the load disturbance (see Fig. 2) can provide consisten and useful CPA information for such classification. These CPIs, therefore, define features of the assessed control system and they are computed from the applied disturbance rejection step response. Some of them are very popular and commonly used in the control literature or by practitioners, e.g. different integral indexes, settling time, maximum overshoot, etc. However, this list also includes 18 additional CPIs that have been defined specifically for this work to facilitate as comprehensive and accurate description of the CS responses to a step load change as possible and by a proxy provide the most relevant and accurate CPA information for the ML algorithms to effectively use.

This section presents the proposed methodology for generating training and validation datasets and assessing selected classification models in terms of their aplicability for the considered CPA problem.

A. Training and validation datasets

AS previously mentioned, the proposed CPA method is based on benchmark disturbance rejection responses determined for normalized SOPDT processes characterised by L_1, L_2 from the assumed ranges and controlled by “optimally” tuned PID controllers. The so called “optimal” tuning is always relative and case-dependent so in this work, it was carried out by solving an optimization problem with constraints. This approach is widely used for deriving tuning rules for various control algorithms, e.g. [22]. In this work, it is assumed that for a certain SOPDT process, the PID tunings are “optimal” if they minimize the integral absolute error (IAE) computed for a disturbance rejection under the constrains defined by the gain and phase margins for a control system assumed as: $A_m \geq 2.5$ and $\phi_m \geq 60^{\circ}$. Such a tuning is rather conservative but acceptable from a practical viewpoint and only used as one of the possible examples. One can easily extend the suggested methodology for a different definitions of the “optimal” PID tuning.

The assumed range of L_1, L_2 variability was covered by a mesh of equidistant points with $\Delta L_1 = \Delta L_2 = 0.1$ so the boundary and internal points of this mesh represent 60 normalized SOPDT processes. For each of them, the “optimal” PID tunings were derived by simulation as described above. Then, based on the spline interpolation between the “optimal” PID tunings determined for neighbouring mesh points, the interpolated “optimal” PID tunings can be calculated for any combination of L_1, L_2 within the assumed ranges.

The “optimal” PID tunings of any considered control system can be modified and corresponding disturbance rejection response can be computed by simulation. The modification was made by multiplying each “optimal” PID parameter by a random number with a normal distribution $N(1, 0.0225)$. Depending on a degree of this modification, one can obtain a control system of acceptable (OK) or not acceptable (NOK) control performance that can be included in training and validation datasets. For each response, all 30 suggested CPIs are computed and their values form a feature vector representing the description of the response of the considered control system (i.e. they form a training sample for the ML algorithms).

Subject to control performance, the binary labelling of each sample as OK or NOK is based on two criteria:

- ± 10% acceptable deviation from the gain and phase margin computed for the control system under consideration, comparing to A_m, ϕ_m values characterizing the benchmark control system for corresponding L_1, L_2.
- Predefined normalized distance e_{dist} between disturbance step responses for the control system under consideration e_{lab} and benchmark e_{bench} for corresponding L_1, L_2:

$$e_{dist} = \frac{\int |e_{bench} - e_{lab}| dt}{\int |e_{bench}| dt}. \tag{1}$$

The control system under consideration is labelled OK if its gain and phase margin fall within the assumed range and $e_{dist} < 0.1$. Otherwise, its is labelled as NOK. This e_{dist} threshold was adjusted experimentally based on preliminary studies which ensures that almost 96% of the control systems that meet this threshold, also meet required gain and phase margins.

The training dataset was generated by selecting 60 000 control systems (samples) determined for random values of pairs L_1, L_2 within their assumed ranges and randomly modified “optimal” PID tunings. It was ensured that for this training dataset, a half of the samples had to be selected from those labelled OK and the other half from the NOK class.

An example of the training dataset with the separation between OK and NOK ranges is graphically presented in Fig. 3 where green dots represent OK cases and red dots – NOK. For clarity, $A_{m,norm}$ and $\phi_{m,norm}$ respectively denote normalized distances of gain and phase margins and thus, acceptable deviation is transformed into [-1, 1] range.

The validation dataset was generated in the same way as training dataset (though completely independently for other random combinations of values of L_1 and L_2 within their ranges) but only 10 000 samples (control systems) for this dataset were selected. It was also ensured that for this dataset, a half had to be selected from those labelled OK and the other half from NOK. A feature vector for each sample was computed in the same way as for the training dataset and its labelling was also based on the same procedure.

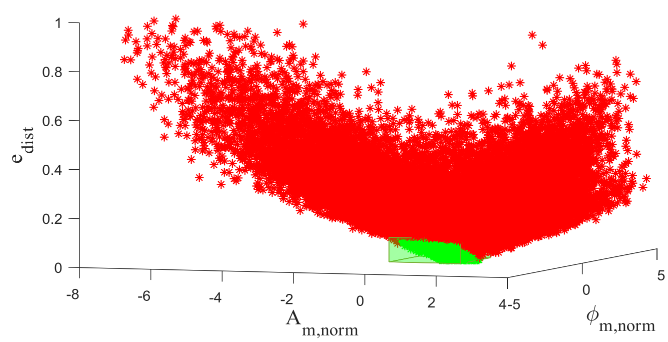

Fig. 3. Graphical representation of exemplary training dataset. OK and NOK performance is marked with green and red colours, respectively. Green box represents assumed range of OK performance.

B. Performance assessment of classification models

Based on the training and validation datasets with 30 CPI features derived as described above, the classification performance of various machine learning algorithms for the considered CPA problem was assessed. Different types of classifiers were selected, ranging from the simple to complex but interpretable models such as Gaussian Naïve Bayes (GNB) [23], Linear Discriminant Analysis (LDA) [24], K-nearest Neighbors (KNN) [25], Decision Tree (DT) [26] and General Fuzzy Min-Max Neural Network trained by an online learning algorithm (Onln-GFMM) [27] or an agglomerative learning algorithm (AGGLO-2) [28], to less transparent but powerful classifiers including kernel-based methods such as Support Vector Machines (SVM) [29] and tree-based ensembles such as Light Gradient Boosted Machine (Light GBM) [30], Extreme Gradient Boosting (XGBoost) [31], Adaptive Boosting (AdaBoost) [32], Extremely Randomized Trees (Extra Trees) [33], and Random Forest (RF) [34]. Apart from GNB and LDA, hyper-parameters of the other models were tuned using random search with the maximum of 100 iterations and 5-fold cross-validation to find the best settings.

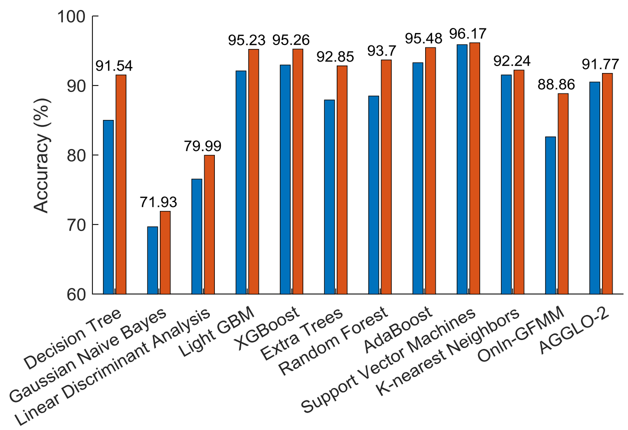

Fig. 4. Classification accuracy for considered classifiers obtained for validation dataset. Comparison between using popular 12 CPI (features) and all 30 CPI (features), both for training and validation.

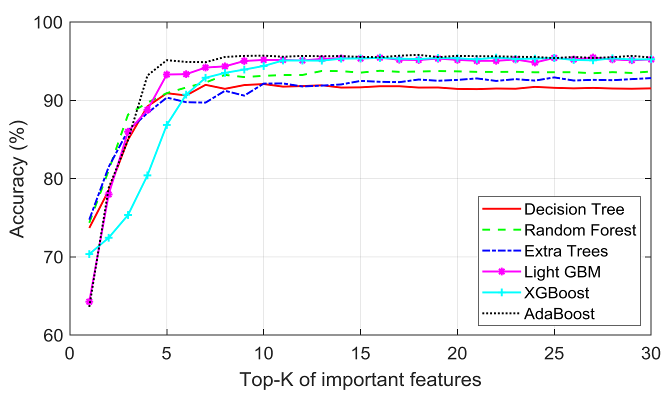

Fig. 5. Accuracy of tree-based learning models on the validation dataset using only top-k of the most important features.

Fig. 4 shows the classification accuracy for these classifiers on the validation dataset. Note that nine models achieved over 91% accuracy, and the best model, i.e., SVM, can achieve more than 96% accuracy. This figure additionaly shows a comparison with the case when training and validation is based only on 12 most popular CPI features. Note that in vast majority of the cases, the classification accuracy drops significantly, which clearly justifies extending the CPI list to the 30 suggested features. As will also be illustrated and discussed later, a suitable combination of a subset of newly introduced and some of the well known CPIs provides the best and most robust discriminative performance for different classifiers.

It can be seen that simple linear classifiers like GNB or LDA cannot reach 80% accuracy on the considered validation dataset. The best performances was observed for other non-linear models. These results indicate the decision boundary between samples of OK and NOK classes are of significantly non-linear nature and cannot be effectively captured by linear decision boundaries of GNB or LDA. As a result, non-linear classifiers were found to be the most appropriate for the CPA classification problem. It can be also noted that the use of complex but interpretable models such as DT, AGGLO-2, or KNN can result in quite good and competitive classification results compared to the other black-box complex models such as SVM or tree-based ensemble models. However, the best performance was usually achieved by using powerful non-linear classifier such as SVM with non-linear kernel and boosted ensemble classifiers, i.e., Light GBM, AdaBoost, and XGBoost.

Although the classification accuracy of fuzzy-based models such as Onln-GFMM and AGGLO-2 was lower than SVM or tree-based ensemble models, a strong argument for the use of these models is that their membership functions can be used to assess how close or far away from the acceptable and non-acceptable control performance boundary each of the classified samples is. This information can be useful to assess the effectiveness of CPA algorithms for monitoring the degradation of controllers in a dynamically changing environment and decide right times to retune the controllers. This opens an interesting research direction for future studies.

For the tree-based models, one of their interesting characteristics is the ability to extract individual CPIs importance scores. Given these importance scores for each tree-based model, the same classifiers were trained using only the top-k of the most important features, with k ranging from 1 to all 30 features. Fig. 5 summarises the accuracy of these tree-based models on different subsets of the top-k of important features.

It can be observed that the accuracy of tree-based learners using from 8 to 15 of the most important features can achieve nearly equal or even better performance on the validation set compared to the case of using all 30 CPI features. This result poses a question of the optimal subset of CPI features which can be used in practice to attain the best classification performance of CPA systems instead of employing all of the proposed features. While noting that substantially smaller set of features can be effectively used, the subsets may be different for different classifiers. Identifying a robust, minimal subset of discriminative features (i.e. CPIs) is out the scope of the current study and will be analysed in more details in the future research.

In the next section, the effectiveness of learning models on simulation and real process data is further assessed and discussed.

IV. Practical Implementation and Verification

This section presents the results of CPA performance based on SVM classifier selected due to its highest accuracy amongst all evaluated classifiers as reported in the previous section.

A. Simulation validation

Initial validation of the proposed CPA system based on the selected SVM classifier was carried out by simulating the control system with SOPDT process defined by (L_1 = 0.4, L_2 = 0.5) and the PID controller. The testing dataset was generated by applying 35 selected tunings of the PID based on the FOPDT approximation of the process step response [35]. Thus, the testing dataset consists of 35 samples, each representing a different PID tuning method. Fig. 6 shows the classification accuracy which for this case is perfect (i.e. 100%).

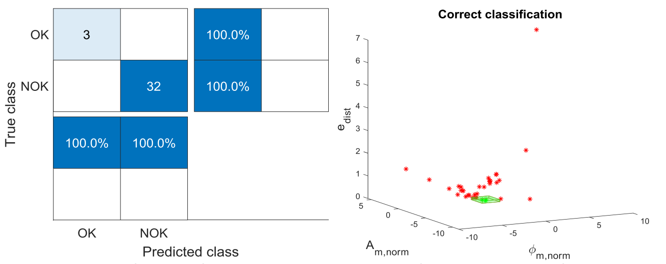

Fig. 6. (Left) Confusion matrix obtained for SVM classifier and test dataset. (Right). Graphical presentation of testing dataset, according to gain and phase margins and e_{dist}. SOPDT process: L_1 = 0.4, L_2 = 0.5.

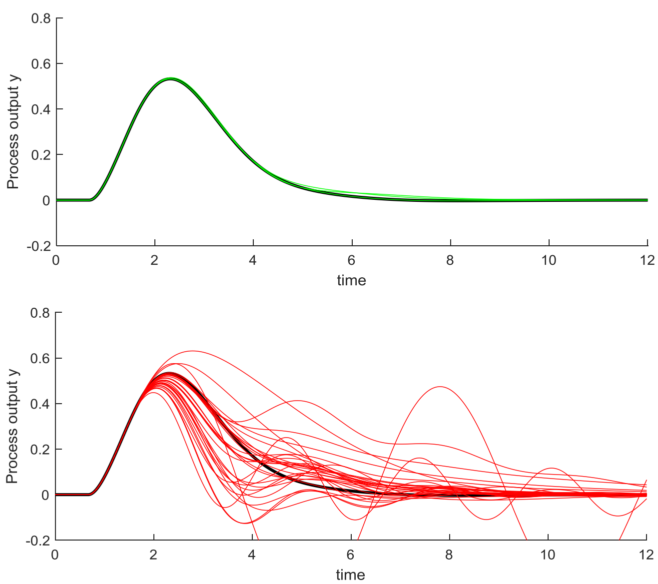

Fig. 7. Comparison of benchmark response (thick, black plot) with testing control systems classified as OK (green upper plots) and NOK (red lower plots). SOPDT (L_1 = 0.4, L_2 = 0.5).

Fig. 7 shows the disturbance rejection responses for each sample from the testing dataset. Note that those classified as OK are very similar to the response of the benchmark control system obtained for considered SOPDT process. At the same time, responses classified as NOK are far from it and some of them are not acceptable in practice.

Second simulation validation was made based on the same methodology but for SOPDT process defined by (L_1 = 0.3, L_2 = 0.9). For this case, one set of tunings results in an unstable behaviour. The classification accuracy shown in Fig. 8 is still very high but not perfect. One control system was misclassified as NOK while in accordance with the labelling methodology described in Section III.A, it should be classified as OK. Fig. 9 shows its disturbance rejection response. However, graphical representation of the test dataset shows that the misclassified sample is very close to the border of NOK region. It is obvious, that in practice, the accuracy of classifiers will not be perfect, especially when testing samples are relatively close to the border between OK and NOK classes. To further distinguish between the cases close to the decision boundaries and provide additional information beyond the class labels, the membership functions of GFMM classifiers can be used and will further be explored in the follow up studies.

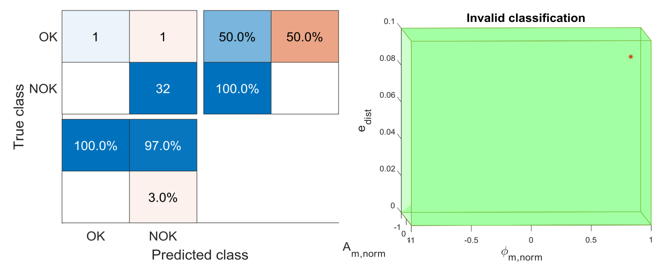

Fig. 8. (Left) Confusion matrix obtained for SVM classifier and test dataset. (Right). Graphical presentation of testing dataset, according to gain and phase margins and e_{dist}. SOPDT (L_1 = 0.3, L_2 = 0.9).

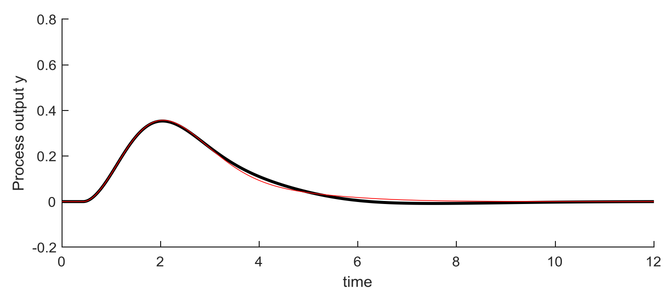

Fig. 9. Comparison of benchmark response (thick, black plot) with testing control system misclassified as NOK (red plot). SOPDT (L_1 = 0.3, L_2 = 0.9).

B. Example of cloud-based practical implementation

The discussed below example of the practical implementation of the CPA system is intended to assess the current control performance of the PID controller implemented in Siemens S7-1500 PLC during its normal operation. This verification should be performed periodically or upon user request to prevent a significant drop in control performance due to slowly varying fluctuations in process dynamics. In order to prevent from excessive computing load required for the CPA functionality, only necessary calculations have been implemented in the control program in PLC in the form of dedicated function block *"ControlPerformanceAssessment"*. Its application jointly with standard PID Compact function block accessible in TIA Portal is shown in Fig. 10. When CPA procedure is ordered, *"InitializeCPA"* is set and *"ControlPerformanceAssessment"* function block waits for the steady state that is detected using the ICM method [36]. Once the steady state has been detected, a load disturbance step change is applied to the process and its amplitude is adjusted to 10% of the range of manipulating variable stored in the structure connected to the *"PID_CompactConfig"* input. Then, the disturbance rejection response data is collected with sampling time defined by *"SamplingTime"* input until the steady state is detected once again by the ICM method after a transient resulting from the process excitation. For monitoring, both steady and transient states are respectively indicated at the outputs *“SteadyState”* and *“TransientState”*. The collected data is stored in PLC’s data memory and when this procedure is completed, the data is sent to OPC server together with the current PID tunings (connected to the input *“PID_CompactCtrlParams”*) using secured OPC UA protocol.

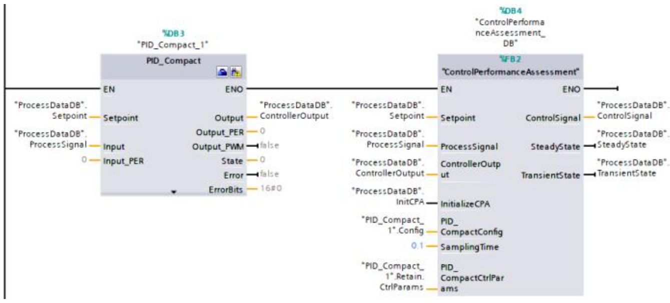

Fig. 10. Siemens S-1500 PLC-based implementation of *“ControlPerformanceAssessment*” function block in control program.

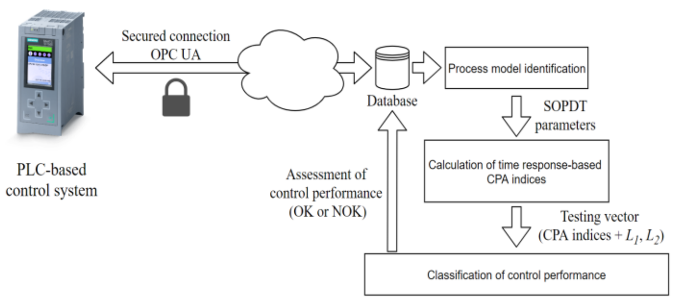

Fig. 11. Architecture of cloud-based implementation of CPA system and its OPC UA connection to PLC-based control system.

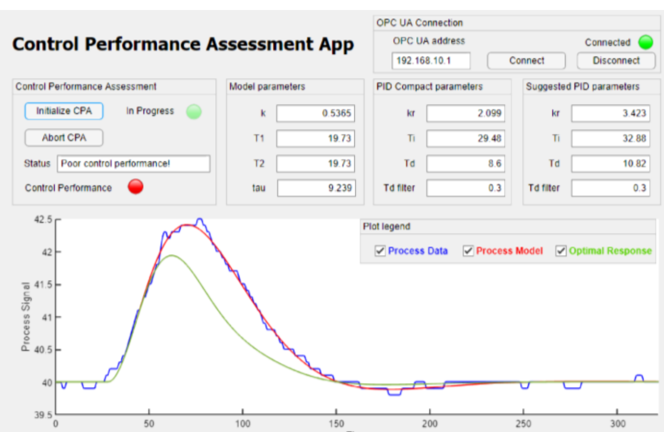

Fig. 12. User interface of exemplary client application for CPA system.

Fig. 11 shows a cloud-based architecture of the considered CPA system. The data collected in PLC is sent to a database and based on this data, SOPDT process parameters are identified by a nonlinear optimization (minimizing of modelling error) procedure (Nelder-Mead simplex algorithm). Then, based on the identified process parameters and the PID tunings, a disturbance rejection response is reconstructed by simulation to minimize the influence of measurement noise. Finally, after computing L_1, L_2 parameters and an appropriate scaling, the CPIs are computed for this simulated response. This is followed by the control performance classification as OK or NOK which is sent to OPC server and then to PLC. It can be also stored in a database and visualised in HMI or SCADA system.

An application of the standard open protocol OPC UA results in full flexibility when it comes to the implementation of client application. An example of the implemented in Matlab client user interface application is presented in Fig. 12. It provides all essential functionalities, such as connection to OPC UA server, initializing CPA procedure, SOPDT model identification and calculating new PID tunings if the performance was classified as NOK. In border cases, the user him/herself can additionally assess the control performance using the graphical visualisation window representing the rejection step response collected from the process and the performance of the benchmark control system.

C. Experimental validation

To further evaluate and strengthen the argument in support of the proposed approach, an experimantal validation was performed based on the part of laboratory heat exchange and distribution plant shown in Fig. 13. Experiments were carried out for the electric flow heater with adjustable heating power P_h within the range 0- 100% of maximal power 12 (kW). The water flows through the heater with the flow rate F and temperature is measured at the heater inlet (T_{in}) and outlet (T_{out}). The control goal is defined to ensure that T_{out}. = T_{SP} (temperature setpoint) by manipulating heating power (manipulating variable). This process exhibits higher (above 2) order dynamics with significant delay time so its dynamical properties are different from the SOPDT used for the training of the CPA system.

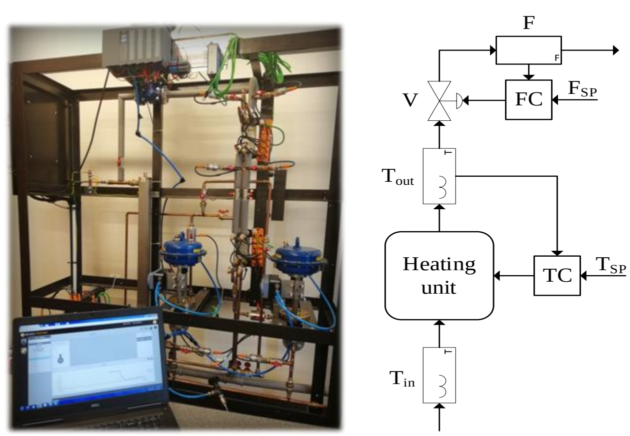

Fig. 13.The overview (left) and simplified diagram (right) of laboratory setup.

For constant flow rate F = 3.5 (L/min), 20 different sets of PID tunings were selected representing 20 different control systems (samples). Some of them were based on well-known tuning methods [35] while the others were adjusted by trial and error method to obtain possibly highest control performance. Then, for each set of the PID tunings, a laboratory setup was operated and the CPA procedure was executed. It was operated in a way described in the subsection IV.B with the applied load disturbance ΔP_h = 10%. The variations of process, benchmark and simulated disturbance rejection resposes obtained during this CPA experiment are shown in the graphical visualisation window in Fig. 12.

The classification for the 20 collected experimental rejection disturbance step responses are shown in Fig. 14. For the visualised measurement data, one can see a presence of the quantization resulting from a limited sensor resolution. Note that in this case, corresponding benchmark responses are slightly different for each assessed control system. It results from the fact that in practice it is impossible to obtain the same results even in the same conditions. Thus, for each CPA experiment, SOPDT approximation of the real disturbance rejection step response was slightly different.

The results show very high classification accuracy for the selected SVM model in the application to CPA of the process exhibiting dynamics more complex than SOPDT. For completeness and comparison, the classification accuracy for the other considered classifiers calculated for the experimental data was investigated and these results show that there are a number of different highly accurate classifiers which could be equally successfully used which indicates the robust character of the underlying CPIs and the overall methodology.

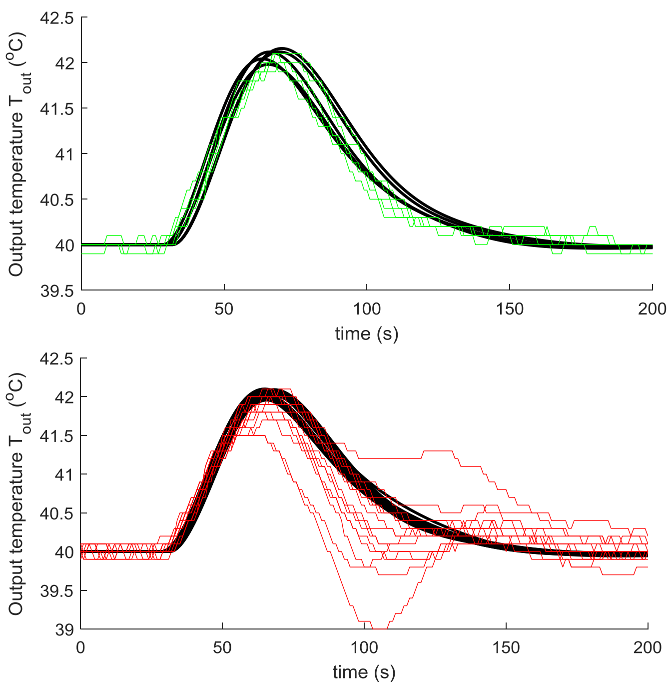

Fig. 14. Comparison of benchmark responses (thick, black plots) with testing control systems classified as OK (green upper plots) and NOK (red lower plots) obtained from laboratory setup.

V. Conclusions

This paper introduced the concept of machine learning (ML) based CPA system and investigated its application to assess the performrmance of PID-based control loops operating processes that exhibit SOPDT dynamics. The proposed concept is based on fusion of up to 30 individual, diverse CPIs computed from the disturbance rejection step response of the assessed control system. These CPIs are used as input features to the ML based classification system. A comparative analysis of a wide range of different machine learning algorithms was presented and important conclusions were drawn in terms of potential reduction of a number of features required for an accurate classification.

CPI features consist of 12 very popular CPIs and 18 additional ones specifically proposed for this study. The classification accuracy and feature importance analysis showed that in general, these additional features provide more effective discriminative, representation of properties of the assessed control systems. Thus, the results indicated that a relatively small subset of them can be used for an accurate assessment of the control performance if a load disturbance step change, required for their calculation, can be executed.

Very promising results showed that this concept can be extended to other classes of control systems, which are based on different (even advanced) controllers operating processes exhibiting different (even more complex) dynamics.The proposed approach, with some indicated extensions forming our future research directions, can be also applied for assessessment of tracking properties of the operating control systems. The included example of practical implementation showed potential applicability and easy transferability of the proposed CPA system into the industiral practice.

References

[1] Z. Yuan, B. Chen, G. Sin, and R. Gani, "State-of-the-art of progress in the optimization-based simultaneous design and control for chemical processes," AIChE Journal, vol. 58, no. 6, pp. 1640–1659, 2012.

[2] T. Klopot, P. Skupin, P. Grelewicz, and J. Czeczot, "Practical PLC-based Implementation of Adaptive Dynamic Matrix Controller for Energy-Efficient Control of Heat Sources," *IEEE Trans. Ind. Electron.*, vol. 0046, no. c, pp. 1–1, 2020, doi: 10.1109/tie.2020.2987272.

[3] P. Nowak, K. Stebel, T. Klopot, J. Czeczot, M. Fratczak, and P. Laszczyk, "Flexible function block for industrial applications of active disturbance rejection controller," *Archives of Control Science*. 28(3), (2018), 349–400.

[4] T. Klopot, P. Skupin, M. Metzger, and P. Grelewicz, "Tuning strategy for dynamic matrix control with reduced horizons," *ISA Trans.*, Mar. 2018, doi: 10.1016/j.isatra.2018.03.003.

[5] M. Fratczak, P. Nowak, T. Klopot, J. Czeczot, S. Bysko and B. Opilski, "Virtual commissioning for the control of the continuous industrial processes – case study," 2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 1032-1037, Międzyzdroje, Poland.

[6] P. Grelewicz, P. Nowak, M. Fratczak, T. Klopot, "Practical Verification of the Advanced Control Algorithms Based on the Virtual Commissioning Methodology – a case study," in *Proc. 23rd International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland, 2018, pp. 217-222.

[7] M. Jelali, Control Performance Management in Industrial Automation: Assessment, Diagnosis and Improvement of Control Loop Performance. Springer-Verlag London, 2013.

[8] T. Samad and A. Annaswamy, "The Impact of Control Technology (2nd edition)," 2014.

[9] P. Van Overschee and B. De Moor, "RAPID: The End of Heuristic PID Tuning," IFAC Proc. Vol., vol. 33, no. 4, pp. 595–600, 2000, doi: 10.1016/s1474-6670(17)38308-8.

[10] W. Caesarendra, T. Wijaya, B. K. Pappachan, and T. Tjahjowidodo, "Adaptation to industry 4.0 using machine learning and cloud computing to improve the conventional method of deburring in aerospace manufacturing industry," Proc. 2019 Int. Conf. Inf. Commun. Technol. Syst. ICTS 2019, pp. 120–124, 2019, doi: 10.1109/ICTS.2019.8850990.

[11] S. K. Panda, A. Blome, L. Wisniewski, and A. Meyer, "IoT Retrofitting Approach for the Food Industry," IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA, vol. 2019-September, pp. 1639–1642, 2019, doi: 10.1109/ETFA.2019.8869093.

[12] C. Zhan, S. Li, and Y. Yang, “Improved process monitoring based on global–local manifold analysis and statistical local approach for industrial process,” J. Process Control, vol. 75, pp. 107–119, 2019, doi: 10.1016/j.jprocont.2018.12.016.

[13] F. Shahni, W. Yu, and B. Young, “Rapid estimation of PID minimum variance,” ISA Trans., vol. 86, pp. 227–237, 2019, doi: 10.1016/j.isatra.2018.10.047.

[14] K. D. Starr, H. Petersen, and M. Bauer, “Control loop performance monitoring – ABB’s experience over two decades,” IFAC-PapersOnLine, vol. 49, no. 7, pp. 526–532, 2016, doi: 10.1016/j.ifacol.2016.07.396.

[15] L. Desborough and R. Miller, “Increasing Customer Value of Industrial Control Performance Monitoring -Honeywell’s Experience,” AIChE Symp. Ser., vol. 98, Jan. 2002.

[16] T. J. Harris, “Assessment of control loop performance,” Can. J. Chem. Eng., vol. 67, no. 5, pp. 856–861, Oct. 1989, doi: 10.1002/cjce.5450670519.

[17] M. J. Grimble, “Controller performance benchmarking and tuning using generalised minimum variance control,” Automatica, vol. 38, no. 12, pp. 2111–2119, 2002, doi: 10.1016/S0005-1098(02)00141-3.

[18] B. S. Ko and T. F. Edgar, “PID control performance assessment: The single-loop case,” AIChE J., vol. 50, no. 6, pp. 1211–1218, 2004, doi: 10.1002/aic.10104.

[19] Z. Liu, H. Y. Su, L. Xie, and Y. Gu, Improved LQG benchmark for control performance assessment on ARMAX model process, vol. 8, no. PART 1. IFAC, 2012.

[20] P. Grelewicz, P. Nowak, J. Czeczot, and M. Fratczak, “Corelation between Conventional and Data-Driven Control Performance Assessment Indices for Heating Process,” in *Proceedings of the 2019 22nd International Conference on Process Control, PC 2019*, 2019, pp. 86–90, doi: 10.1109/PC.2019.8815041.

[21] Z. Gao, “Scaling and Bandwidth-Parameterization based Controller Tuning,” in *Proceedings of the American Control Conference*, 2003, vol. 6, pp. 4989–4996, doi: 10.1109/acc.2003.1242516.

[22] P. Nowak, J. Czeczot, and T. Klopot, “Robust tuning of a first order reduced Active Disturbance Rejection Controller,” *Control Eng. Pract.*, 2018, doi: 10.1016/j.conengprac.2018.02.001.

[23] H. Zhang, “The optimality of naive bayes,” in *Proc. of the 17th International Florida Artificial Intelligence Research Society Conference*, 2004, p. 562–567.

[24] J. Ye, “Least squares linear discriminant analysis,” in *Proc. of the 24th international conference on Machine learning*, 2007, pp. 1087– 1093.

[25] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[26] L. Breiman, J. Friedman, R. Olshen, and C. Stone, “Classification and Regression Trees,” Wadsworth, Belmont, CA, 1984.

[27] B. Gabrys and A. Bargiela, “General fuzzy min-max neural network for clustering and classification,” *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 769–783, 2000.

[28] B. Gabrys, “Agglomerative learning algorithms for general fuzzy minmax neural network,” *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 32, no. 1, pp. 67–82, 2002.

[29] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems* 30, 2017, pp. 3146–3154.

[31] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[32] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119-139, 1997.

[33] P. Geurts, D. Ernst., and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, 63(1), 3-42, 2006.

[34] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] A. O’Dwyer, *Handbook of PI and PID Controller Tuning Rules*. IMPERIAL COLLEGE PRESS, 2009.

[36] P. Grelewicz, P. Nowak, J. Czeczot, and J. Musial, “Increment Count Method and its PLC-based Implementation for Autotuning of Reduced-Order ADRC with Smith Predictor,” *IEEE Trans. Ind. Electron.*, (accepted for publication)

Patryk Grelewicz received the M.S. degree in automatic control from Silesian University of Technology, Gliwice, Poland in 2018.

He is currently pursuing the Ph.D. degree in automatic control at the same university. His research interest includes application of advanced control strategies and control performance assessment algorithms for industrial dynamical systems.

Thanh Tung Khuat received the B.E degree in Software Engineering from University of Science and Technology, Danang, Vietnam, in 2014.

He is currently working towards the Ph.D. degree in machine learning at University of Technology Sydney, Ultimo, NSW, Australia. His research interests include machine learning, fuzzy systems, knowledge discovery, evolutionary computation, and intelligent optimisation techniques.

Jacek Czeczot received the M.S. degree in computer control systems from Silesian University of Technology, in 1994 and the Ph.D. degree in automatic control and robotics from the same university, in 1997.

He is Professor with the Faculty of Automatic Control, Electronics and Computer Science at Silesian University of Technology in Gliwice, Poland. His research interests include mathematical modeling and advanced model-based control of industrial dynamical systems.

Tomasz Kłopot received the M.S. degree in computer control systems from Silesian University of Technology, in 2002 and the Ph.D. degree in automatic control and robotics from the same university, in 2007.

He is Assistant Professor with the Faculty of Automatic Control, Electronics and Computer Science at Silesian University of Technology in Gliwice, Poland. His research interests include advanced control algorithms and industrial control networks.

Bogdan Gabrys received the M.Sc. degree in electronics and telecommunication from Silesian Technical University, Gliwice, Poland, in 1994, and the Ph.D. degree in computer science from Nottingham Trent University, Nottingham, U.K., in 1998.

He is currently a Professor of Data Science at the University of Technology Sydney, NSW, Australia. His research activities have concentrated on the areas of data science, complex adaptive systems, computational intelligence, machine learning, predictive analytics, and their diverse applications.

Supplemental Material for the Paper: Application of Machine Learning to Performance Assessment for a class of PID-based Control Systems

Patryk Grelewicz, Thanh Tung Khuat, Jacek Czeczot, Tomasz Klopot, Bogdan Gabrys

I. Complete list of Used Control Performance Indices (CPIs)

The complete list of used CPIs with their short descriptions is presented in Table S.I. The most popular CPIs that are frequently used as control performance measures are highlighted with grey colour while the other CPIs are defined specifically for this work.

TABLE S.I
THE COMPLETE LIST OF CPIS

Control Performance Index	Short description	Acronym
MaxPeak	Maximum value of dynamic system response	F1
MaxPeakTime	The moment, when the maximum peak occurs	F2
MinPeak	Minimum value of dynamic system response, absolute value	F3
MinPeakTime	The moment, when the minimum peak occurs	F4
MinToMax	The ratio of minimum and maximum peak	F5
MaxToMinTime	The difference of time, when maximum and minimum peaks occur $MaxToMinTime = MinPeakTime - MaxPeakTime$	F6
SettlingTime	The moment, when the response of system is within the range of 1% of its steady state $\|e\| < 0.01$	F7
IAE	Integral Absolute Error $IAE = \int \|e\| dt$	F8
ISE	Integral Square Error $ISE = \int e^2 dt$	F9
ITAE	Integral Time Absolute Error $ITAE = \int t\|e\|dt$	F10
IT2AE	Integral Time Square Absolute Error $IT2AE = \int t^2 \|e\| dt$	F11
IAEPos	Integral Absolute Error calculated for positive values of system response $IAEPos = \int \|e\|dt, e > 0$	F12
IAENeg	Integral Absolute Error calculated for negative values of system response $IAENeg = \int \|e\|dt, e < 0$	F13
IAENegToPos	Ratio of *IAENeg* and *IAEPos*	F14
DecayRatio	Ratio of maximum peak to second positive peak $DecayRatio = \frac{2^{nd}Peak}{MaxPeak}$	F15
DecayRatioTime	The difference between time, when maximum and second peaks appeared $DecayRatioTime = 2^{nd}PeakTime - MaxPeakTime$	F16
PeakSettlingTime	Difference between *SettlingTime* and *MaxPeakTime*	F17
TimePos	The total amount of time, when the response of the system is positive $TimePos = \int t\, dt, e > 0$	F18
TimeNeg	The total amount of time, when the response of the system is negative $TimeNeg = \int t\, dt, e < 0$	F19
TimeNegToPos	The ratio of *TimeNeg* and *TimePos*	F20
RisingTime	Rising time of the maximum peak, calculated as a time of reaching from 5% to 95% of *MaxPeak*	F21
FallingTime	Falling time of the maximum peak, calculated as a time of reaching from 95% to 5% of *MaxPeak*	F22
RisingToFallingTime	Ratio of *RisingTime* and *FallingTime*	F23
25%DistRejected	The moment, when the response of system is within the range of 25% of *MaxPeak*, $\|e\| < 25\% * MaxPeak$	F24
50%DistRejected	The moment, when the response of system is within the range of 50% of *MaxPeak*, $\|e\| < 50\% * MaxPeak$	F25
75%DistRejected	The moment, when the response of system is within the range of 75% of *MaxPeak*, $\|e\| < 75\% * MaxPeak$	F26
ZeroCrossingTime	The first moment, when the response of the system crosses the zero value	F27
MaxDiff	Maximum value of the derivative of the dynamic response	F28
MinDiff	Minimum value of the derivative of the dynamic response, absolute value	F29
DiffMaxToMin	Ratio of *MaxDiff* and *MinDiff*	F30

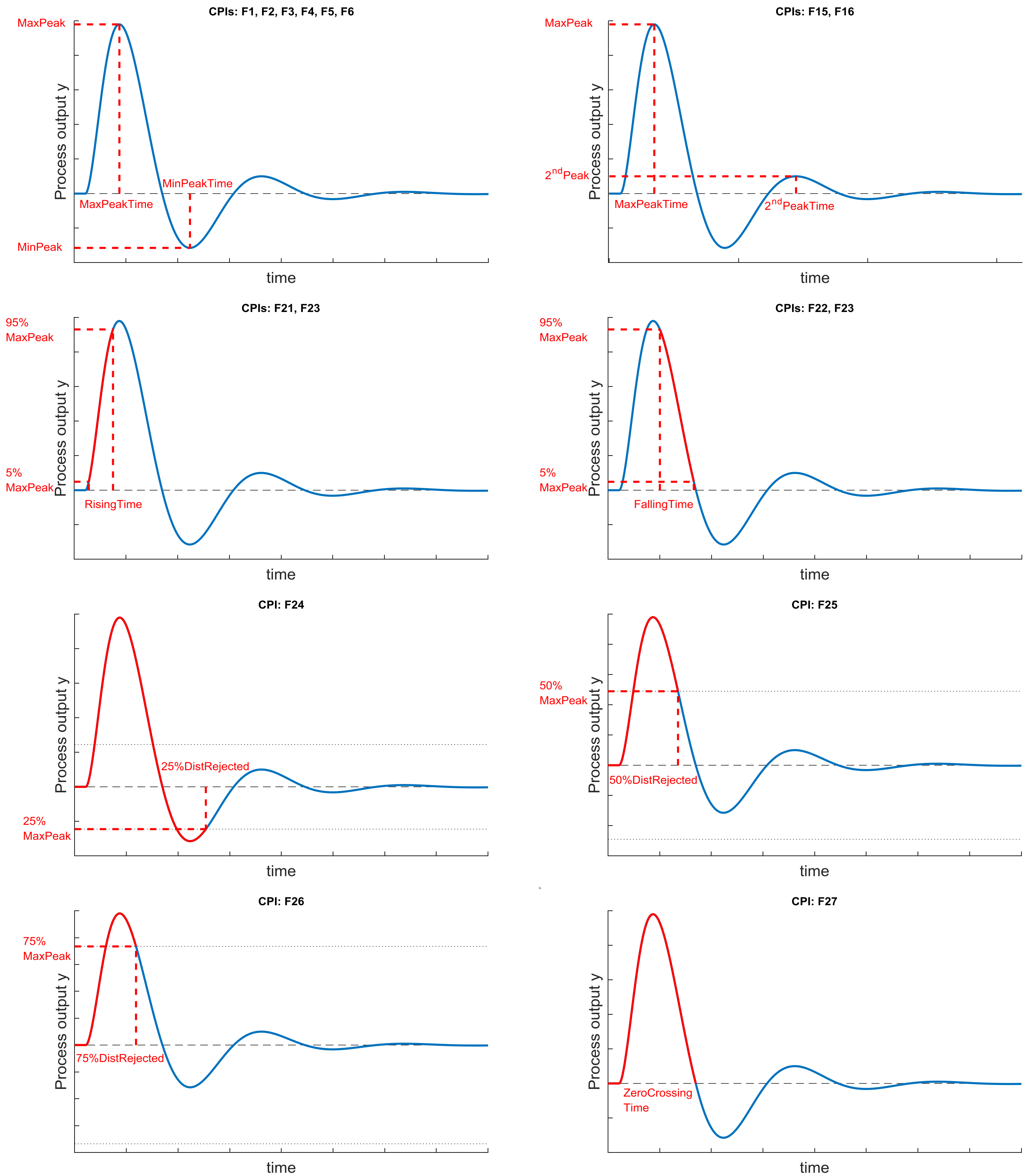

Fig. S1 Graphical interpretation of a set of chosen CPIs: *MaxPeak*, *MaxPeakTime*, *MinPeak*, *MinPeakTime*, *2ndPeak* (for calculating *DecayRatio*), *2ndPeakTime* (for calculating *DecayRatioTime*), *RisingTime*, *FallingTime*, *25%DistRejected*, *50%DistRejected*, *75%DistRejected*, *ZeroCrossingTime*.

II. Hyperparameter optimization

The parameters of studied classification methods were obtained using a hyperparameter optimization approach described in the main manuscript. The results are presented in Table S.II, including the considered range and optimal value of each hyperparameter.

TABLE S.II
HYPERPARAMETER OPTIMIZATION RESULTS FOR STUDIED CLASSIFICATION ALGORITHMS

Classification algorithm	Parameter	Range	Optimal value
Decision Trees	Max depth	[4, 20]	19
	Min samples per leaf	[4, 30]	4
Light GBM	Max depth	[4, 20]	20
	Min samples per leaf	[4, 30]	12
	Sampling rate	{0.3, 0.4, 0.5, 0.6, 0.7}	0.4
	% features used	{20%, 30%, 40%, 50%, 60%, 70%}	70%
	Learning rate	{0.025, 0.05, 0.1, 0.2, 0.3}	0.3
	No of estimators	{30, 50, 70, 100, 150, 200}	200
XGBoost	Max depth	[4, 20]	8
	Sampling rate	{0.3, 0.4, 0.5, 0.6, 0.7}	0.7
	% features used	{20%, 30%, 40%, 50%, 60%, 70%}	70%
	Learning rate	{0.025, 0.05, 0.1, 0.2, 0.3}	0.2
	Gamma	{0, 0.1, 0.2, 0.3, 0.4, 1, 1.5, 2}	1
	No of estimators	{30, 50, 70, 100, 150, 200}	200
Extra Trees	Max depth	[4, 20]	20
	Min samples per leaf	[4, 30]	6
	% features used	{20%, 30%, 40%, 50%, 60%, 70%}	40%
	Sampling rate	{0.3, 0.4, 0.5, 0.6, 0.7}	0.7
	No of estimators	{30, 50, 70, 100, 150, 200}	50
Random Forest	Max depth	[4, 20]	20
	Min samples per leaf	[4, 30]	6
	% features used	{20%, 30%, 40%, 50%, 60%, 70%}	40%
	Sampling rate	{0.3, 0.4, 0.5, 0.6, 0.7}	0.7
	No of estimators	{30, 50, 70, 100, 150, 200}	50
AdaBoost	Max depth	[4, 20]	11
	Min samples per leaf	[4, 30]	12
	No of estimators	{30, 50, 70, 100, 150, 200}	150
	Learning rate	{0.001, 0.01, 0.1, 0.2, 0.5, 1}	0.1
Support Vector Machines	Kernel	{'rbf', 'sigmoid', 'linear'}	rbf
	Gamma	{2^-15, 2^-13, …, 2^3}	8
	C	{2^-5, 2^-3, …, 2^15}	512
K-nearest Neighbour	K	{1, 3, …, 29}	5
Onln-GFMM	Maximum hyperbox size θ	{0.1, 0.15, …, 0.55, 0.6}	0.1
AGGLO-2	Maximum hyperbox size θ	{0.1, 0.15, …, 0.55, 0.6}	0.4

III. Possibility of Feature Reduction

To check the possibility of feature reduction, correlation coefficients (Table S.III) and feature importance for tree based models (Table S.IV) were calculated. The highly correlated groups of indices were colour-coded in Table S.III and Table S.IV. One can notice that the most important features in vast majority of cases are the representatives of obtained colour-coded groups. What is more, the classification accuracy does not increase, when the number of features is higher than approximately 10 (Fig. 5). These results suggest, that the number of effective CPIs can be reduced without any significant drop in classification accuracy. This issue will be studied in the future, as with a small number of relatively easily computable features, the overall computational complexity decreases and a type of the CPA system proposed in this work can be implemented directly in PLC, as a ready-to-use general-purpose function block.

TABLE S.III

Correlation coefficients calculated for each pair of CPI

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30
F1	1.000	0.902	0.641	0.863	0.790	0.779	0.725	0.946	0.931	0.875	0.810	0.948	0.341	0.789	0.712	0.550	0.389	0.703	0.530	0.176	0.828	0.767	0.736	0.863	0.873	0.885	0.882	0.415	0.807	0.823
F2		1.000	0.468	0.937	0.620	0.832	0.890	0.986	0.951	0.972	0.940	0.986	0.172	0.624	0.548	0.451	0.585	0.873	0.768	0.032	0.985	0.888	0.901	0.994	0.997	0.999	0.954	0.016	0.485	0.829
F3			1.000	0.566	0.925	0.592	0.119	0.496	0.433	0.394	0.326	0.508	0.894	0.936	0.877	0.362	0.255	0.344	0.089	0.538	0.428	0.498	0.219	0.417	0.423	0.443	0.600	0.513	0.772	0.675
F4				1.000	0.692	0.973	0.852	0.899	0.820	0.835	0.772	0.900	0.258	0.693	0.623	0.532	0.580	0.936	0.607	0.277	0.948	0.934	0.807	0.938	0.935	0.936	0.996	0.006	0.509	0.915
F5					1.000	0.692	0.330	0.651	0.586	0.543	0.470	0.659	0.684	0.999	0.978	0.488	0.033	0.452	0.154	0.357	0.569	0.666	0.307	0.581	0.578	0.594	0.718	0.551	0.846	0.785
F6						1.000	0.769	0.779	0.677	0.688	0.608	0.781	0.298	0.691	0.630	0.549	0.538	0.913	0.459	0.419	0.859	0.901	0.690	0.837	0.829	0.830	0.955	0.020	0.490	0.909
F7							1.000	0.847	0.806	0.840	0.813	0.841	0.200	0.327	0.272	0.402	0.890	0.893	0.832	0.048	0.906	0.837	0.848	0.915	0.910	0.902	0.843	0.206	0.236	0.697
F8								1.000	0.986	0.981	0.947	1.000	0.203	0.654	0.575	0.453	0.522	0.801	0.726	0.017	0.942	0.826	0.881	0.965	0.974	0.980	0.920	0.159	0.584	0.792
F9									1.000	0.981	0.955	0.985	0.158	0.591	0.509	0.408	0.484	0.714	0.721	0.068	0.887	0.728	0.878	0.921	0.936	0.943	0.845	0.212	0.581	0.694
F10										1.000	0.991	0.980	0.133	0.550	0.471	0.341	0.524	0.767	0.787	0.102	0.934	0.776	0.904	0.956	0.966	0.969	0.864	0.067	0.458	0.700
F11											1.000	0.945	0.091	0.478	0.403	0.257	0.507	0.723	0.811	0.180	0.908	0.729	0.890	0.928	0.939	0.940	0.806	0.013	0.373	0.629
F12												1.000	0.217	0.663	0.582	0.453	0.512	0.800	0.718	0.026	0.942	0.826	0.878	0.964	0.973	0.979	0.922	0.165	0.591	0.795
F13													1.000	0.704	0.627	0.090	0.527	0.073	0.394	0.600	0.139	0.159	0.030	0.107	0.123	0.146	0.303	0.450	0.579	0.395
F14														1.000	0.974	0.478	0.041	0.454	0.152	0.363	0.574	0.665	0.317	0.585	0.583	0.599	0.721	0.543	0.840	0.783
F15															1.000	0.469	0.063	0.364	0.143	0.250	0.503	0.625	0.219	0.517	0.509	0.523	0.645	0.535	0.785	0.724
F16																1.000	0.265	0.434	0.288	0.175	0.434	0.524	0.331	0.447	0.440	0.443	0.511	0.167	0.475	0.588
F17																	1.000	0.717	0.715	0.117	0.629	0.603	0.609	0.635	0.624	0.607	0.548	0.382	0.064	0.413
F18																		1.000	0.649	0.303	0.920	0.892	0.816	0.897	0.890	0.884	0.925	0.256	0.267	0.823
F19																			1.000	0.502	0.787	0.712	0.728	0.811	0.799	0.785	0.614	0.329	0.005	0.478
F20																				1.000	0.045	0.126	0.023	0.001	0.009	0.020	0.265	0.216	0.390	0.343
F21																					1.000	0.924	0.898	0.994	0.992	0.989	0.959	0.136	0.366	0.842
F22																						1.000	0.679	0.914	0.897	0.892	0.937	0.105	0.396	0.903
F23																							1.000	0.893	0.908	0.908	0.814	0.207	0.245	0.649
F24																								1.000	0.999	0.997	0.950	0.064	0.415	0.829
F25																									1.000	0.999	0.948	0.046	0.429	0.818
F26																										1.000	0.951	0.022	0.451	0.822
F27																											1.000	0.021	0.533	0.920
F28																												1.000	0.823	0.057
F29																													1.000	0.600
F30																														1.000

TABLE S.IV

THE RANK OF CPI FEATURES AND ACCURACY (%) OF TREE-BASED MODELS ON THE VALIDATION DATASET USING TOP-K OF THE MOST IMPORTANT FEATURES

Rank	Decision Tree		Random Forest		Extra Trees		Light GBM		XGBoost		AdaBoost	
	Feature	Accuracy	Feature	Accuracy	Feature	Accuracy	Feature	Accuracy	Feature	Accuracy	Feature	Accuracy
1	F23	73.70	F23	74.36	F23	74.76	F30	64.28	F13	70.36	F30	63.61
2	F3	78.50	F30	81.06	F30	81.5	F23	77.99	F3	72.45	F23	78.82
3	F30	84.98	F3	88.19	F3	86.15	F29	86	F22	75.36	F29	84.95
4	F22	89.23	F13	89.72	F22	88.38	F1	88.78	F17	80.42	F1	93.18
5	F29	90.92	F17	90.93	F17	90.36	F28	93.31	F23	86.87	F20	95.14
6	F28	90.63	F22	91.65	F19	89.77	F9	93.35	F30	90.74	F9	94.92
7	F1	91.99	F15	92.28	F14	89.72	F20	94.2	F5	92.89	F28	94.88
8	F19	91.48	F29	93.26	F20	91.23	F22	94.35	F12	93.52	F3	95.55
9	F15	91.96	F19	92.98	F13	90.61	F3	95.04	F26	93.94	F14	95.69
10	F13	92.09	F28	93.14	F29	92.15	F14	95.17	F15	94.4	F17	95.72
11	F5	91.76	F5	93.24	F5	92.17	F5	95.17	F29	95.1	F19	95.58
12	F9	91.83	F20	93.25	F6	91.78	F19	95.11	F1	95.1	F5	95.68
13	F20	91.93	F1	93.76	F4	91.89	F15	95.3	F20	95.08	F15	95.64
14	F17	91.64	F16	93.74	F1	92.04	F16	95.39	F14	95.33	F13	95.66
15	F14	91.67	F14	93.55	F27	92.49	F6	95.35	F19	95.37	F12	95.56
16	F16	91.81	F9	93.79	F16	92.38	F2	95.46	F2	95.43	F18	95.51
17	F24	91.81	F8	93.65	F28	92.33	F17	95.2	F8	95.33	F22	95.71
18	F11	91.64	F12	93.69	F9	92.67	F18	95.17	F16	95.29	F16	95.82
19	F12	91.65	F6	93.75	F15	92.51	F12	95.34	F28	95.41	F8	95.53
20	F6	91.47	F7	93.71	F2	92.62	F13	95.18	F6	95.38	F6	95.69
21	F18	91.44	F2	93.65	F8	92.81	F8	95.06	F9	95.25	F27	95.65
22	F8	91.52	F26	93.64	F18	92.49	F26	95.06	F10	95.47	F4	95.63
23	F25	91.49	F10	93.66	F10	92.72	F7	95.23	F21	95.3	F7	95.58
24	F21	91.73	F18	93.59	F26	92.53	F21	94.84	F25	95.34	F11	95.55
25	F7	91.61	F24	93.6	F12	92.94	F27	95.43	F27	95.42	F24	95.41
26	F10	91.54	F25	93.6	F24	92.54	F11	95.23	F18	95.17	F25	95.55
27	F2	91.6	F27	93.47	F7	92.63	F24	95.48	F7	95.12	F10	95.43
28	F4	91.52	F21	93.61	F21	92.58	F25	95.24	F24	95.38	F2	95.52
29	F26	91.49	F11	93.51	F25	92.72	F4	95.13	F11	95.23	F21	95.69
30	F27	91.54	F4	93.7	F11	92.85	F10	95.23	F4	95.26	F26	95.48

IV. CLASSIFICATION ACCURACY FOR SIMULATION AND EXPERIMENTAL PROCESS DATASETS

The studied classifiers were tested on both simulation sets (for $L_1 = 0.4$, $L_2 = 0.5$ and $L_1 = 0.3$, $L_2 = 0.9$) and on real process dataset. In case of real process dataset, each response was labelled based on the labelling method suggested in the paper in Section III.A and its SOPDT approximation. The obtained accuracies are generally very high and similar to the results obtained for the validation dataset (Fig. 4).

TABLE S.V

CLASSIFICATION ACCURACY (%) FOR SIMULATION AND EXPERIMENTAL PROCESS DATASETS

Classification algorithm	Simulation dataset $L_1 = 0.4$, $L_2 = 0.5$		Simulation dataset $L_1 = 0.3$, $L_2 = 0.9$		Real process dataset	
	Confusion matrix	Accuracy	Confusion matrix	Accuracy	Confusion matrix	Accuracy
Decision Trees	$\begin{bmatrix}3 & 0\\0 & 32\end{bmatrix}$	100	$\begin{bmatrix}1 & 1\\2 & 30\end{bmatrix}$	91.17	$\begin{bmatrix}5 & 1\\0 & 14\end{bmatrix}$	95
Gaussian Naïve Bayes	$\begin{bmatrix}2 & 1\\3 & 29\end{bmatrix}$	88.57	$\begin{bmatrix}1 & 1\\5 & 27\end{bmatrix}$	82.35	$\begin{bmatrix}6 & 0\\3 & 11\end{bmatrix}$	85
Linear Discriminant Analysis	$\begin{bmatrix}1 & 2\\3 & 29\end{bmatrix}$	85.71	$\begin{bmatrix}1 & 1\\1 & 31\end{bmatrix}$	94.11	$\begin{bmatrix}6 & 0\\3 & 11\end{bmatrix}$	85
Light GBM	$\begin{bmatrix}3 & 0\\1 & 31\end{bmatrix}$	97.14	$\begin{bmatrix}1 & 1\\0 & 32\end{bmatrix}$	97.05	$\begin{bmatrix}5 & 1\\0 & 14\end{bmatrix}$	95
XGBoost	$\begin{bmatrix}3 & 0\\1 & 31\end{bmatrix}$	97.14	$\begin{bmatrix}1 & 1\\0 & 32\end{bmatrix}$	97.05	$\begin{bmatrix}6 & 0\\0 & 14\end{bmatrix}$	100
Extra tree	$\begin{bmatrix}2 & 1\\0 & 32\end{bmatrix}$	97.14	$\begin{bmatrix}1 & 1\\1 & 31\end{bmatrix}$	94.11	$\begin{bmatrix}5 & 1\\0 & 14\end{bmatrix}$	95
Random Forest	$\begin{bmatrix}3 & 0\\0 & 32\end{bmatrix}$	100	$\begin{bmatrix}1 & 1\\0 & 32\end{bmatrix}$	97.05	$\begin{bmatrix}5 & 1\\0 & 14\end{bmatrix}$	95
AdaBoost	$\begin{bmatrix}3 & 0\\0 & 32\end{bmatrix}$	100	$\begin{bmatrix}1 & 1\\0 & 32\end{bmatrix}$	97.05	$\begin{bmatrix}6 & 0\\0 & 14\end{bmatrix}$	100
Support Vector Machine	$\begin{bmatrix}3 & 0\\0 & 32\end{bmatrix}$	100	$\begin{bmatrix}1 & 1\\0 & 32\end{bmatrix}$	97.05	$\begin{bmatrix}5 & 1\\0 & 14\end{bmatrix}$	95
k-Nearest Neighbour	$\begin{bmatrix}3 & 0\\1 & 31\end{bmatrix}$	97.14	$\begin{bmatrix}1 & 1\\0 & 32\end{bmatrix}$	97.05	$\begin{bmatrix}6 & 0\\0 & 14\end{bmatrix}$	100
Onln-GFMM	$\begin{bmatrix}3 & 0\\1 & 31\end{bmatrix}$	97.14	$\begin{bmatrix}1 & 1\\1 & 31\end{bmatrix}$	94.11	$\begin{bmatrix}3 & 3\\0 & 14\end{bmatrix}$	85
AGGLO-2	$\begin{bmatrix}2 & 1\\0 & 32\end{bmatrix}$	97.14	$\begin{bmatrix}1 & 1\\0 & 32\end{bmatrix}$	97.05	$\begin{bmatrix}4 & 2\\0 & 14\end{bmatrix}$	90