

# AI for Closed-Loop Control Systems

New Opportunities for Modeling, Designing, and Tuning Control Systems

JULIUS SCHÖNING, Osnabrück University of Applied Sciences, Germany

ADRIAN RIECHMANN, Science to Business GmbH, Germany

HANS-JÜRGEN PFISTERER, Osnabrück University of Applied Sciences, Germany

Control Systems, particularly closed-loop control systems (CLCS), are frequently used in production machines, vehicles, and robots nowadays. CLCS are needed to actively align actual values of a process to a given reference or set values in real-time with a very high precession. Yet, artificial intelligence (AI) is not used to model, design, optimize, and tune CLCS. This paper will highlight potential AI-empowered and -based control system designs and designing procedures, gathering new opportunities and research direction in the field of control system engineering. Therefore, this paper illustrates which building blocks within the standard block diagram of CLCS can be replaced by AI, i.e., artificial neuronal networks (ANN). Having processes with real-time contains and functional safety in mind, it is discussed if AI-based controller blocks can cope with these demands. By concluding the paper, the pros and cons of AI-empowered as well as -based CLCS designs are discussed, and possible research directions for introducing AI in the domain of control system engineering are given.

CCS Concepts: • Computing methodologies → Computational control theory; Artificial intelligence;  
• Information systems → Process control systems.

Additional Key Words and Phrases: Closed-Loop Control Systems, Adaptive Control Systems, Artificial Intelligent, Artificial Neuronal Networks, Real-Time, Functional Safety

## 1 INTRODUCTION

In an ideal world, the controller of CLCS sets the control variables of processes or systems so that the system response perfectly matches the exact reference value. The intended process response is strictly adhered to the reference value—at every time, no matter how high the disturbance on the process is. However, real-world processes and systems such as mechatronical actuators within vehicles, bio-chemical processes within reactors, and temperature regulation systems within buildings can not be controlled without any over as well as under shootings, settling times, and oscillations. Thus, every technology bringing real-world controllers closer to the theoretical ideal controllers is welcome. Admittedly, even with nowadays computational power, the general procedure of designing control systems [6, 26] has not changed within the last four decades.

Introducing AI in the domain of closed-loop and feedback control systems, hereinafter referred to as CLCS, will improve the entire CLCS and the controller behavior, especially for multivariable and nonlinear processes. Further AI might be capable of predicting possible disturbances even before they affect the process. It is conceivable that AI will replace the conventional controller block entirely and directly interpret the sensor values as process feedback. As a result, the CLCS block diagrams and the controller design procedure will adopt, enabling AI within control system engineering.

In the remainder of this paper, the state of the art concerning CLCS and physics-informed AI are summarized in Section 2. Discussing new AI-empowered and -based block diagrams for CLCS, Section 3 will introduce new blocks and new design patterns in detail. Focusing on functional safety, the most critical property of CLCS for specific processes, Section 4 will show that AI-based CLCS and safety will go together. By concluding this paper, Section 5 the pros and cons of AI-empowered as well as -based CLCS designs are discussed and

possible research directions for the usage of AI within CLCS are highlighted. In this paper, for simplification, the term process stands for both: the controlled process and the controlled systems. Whereas the term system expresses the entire open- or closed-loop system, hereafter.

## 2 STATE OF THE ART

AI and especially ANN have not yet made significant inroads into the field of control systems, including both open- as well as closed-loop systems. Figure 1 prove this assessment by comparing the different curve gradients of the publication count with the keyword *Neural Network* in contrast to the publication count with the keyword combination *Closed-Loop Control Systems* and *Neural Network*. Some reasons for the, so far, non-existing combination of control system engineering and AI can be the following:

- the complexity of the phase space described by the differential equations,
- the unpredictability of the disturbances on the process to be controlled, as well as
- the tradition in control system engineering, which is characterized by describing the process in its entirety in physical models.

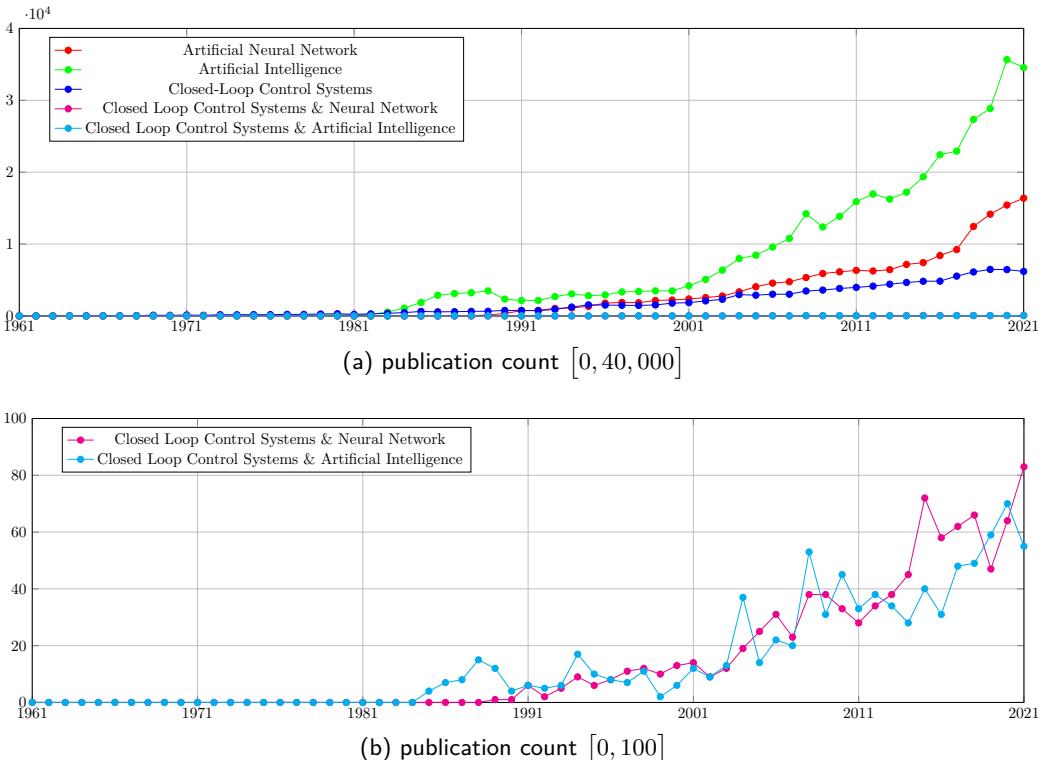


Fig. 1. Count of publication with the keywords: *Closed-Loop Control Systems*, *Artificial Neuronal Network*, *Artificial Intelligence* and the keyword combinations: *Closed-Loop Control Systems & Neuronal Network*, *Closed-Loop Control Systems & Artificial Intelligent*; calculated from the Elsevier Scopus bibliographic database [8].

## 2.1 Classical Closed Loop Control Systems

In CLCS, the most common controller used is the proportional– integral–derivative (PID) controller. This high usage is because most engineers understand how to vary the three parameters [3], based on the past (I), present (P), and future (D) control error [1] for achieving the intended CLCS behavior. However, PID controllers do not work well for nonlinear processes as well as for processes with small dead-times [3]. Further, depending on its parameter, PID controllers are prone to over- and undershot or reach the set-point very slow [28]. As countermeasures, the parameter tuning strategies [28] have been optimized, and the single PID controller design has been decomposition more complex designs such as PID-P, PI-PD, and PI-D controller [3].

Nonlinear processes with uncertainties distributed non-periodic values are still hard to control [1]. Processes with multivariable responses and reference are also unfeasible to control by conventional CLCS design.

## 2.2 AI-Based Closed Loop Control Systems

The gap between the non-existing AI and control system engineering combination was recently recognized, so the first scientific papers appeared. For example, Rackauckas et al. [21] were recently able to present a general approach using ANN to approximate differential equations, even finding missing terms. The developed methodology of universal differential equations discovery can approximate the differential equations of a specific process based on its recorded data.

For combining AI and CLCS, differentiable programming languages like Julia [4] are needed to solve the differential equations and train the ANN within the same software code. The benefit of Julia as a programming language is shown inter alia by Rackauckas et al. [20]. The concept of physics-informed neural networks [22] is based on the same conceptual idea but uses different programming languages to bring differential equations and ANN together. Still to be mentioned are the two more review papers on nonlinear [23] and dynamic [5] differential equations and ANN, respectively. In engineering applications, the gap between control engineering and AI currently continues to exist.

## 2.3 Procedure of the Control System Design

Starting at the turn of the millennium, an increasing amount of computer-aided applied control designing software solutions like Simulink (MATLAB) [29], Xcos (Scilab) [9], and control (Octave) [19] occurs and manifests the currently used best practices in the procedure of the controller design. As illustrated in Figure 2, 14 steps are common to get from the process needed to control to the final control system design [26]. Note, in Figure 2 an additional 15. step is added due to personal experience.

## 3 AI EMPOWERMENT FOR CLOSED-LOOP CONTROL SYSTEMS

Considering the control system design procedure and the CLCS block diagram, AI can be applied in different design steps and blocks. Thus, this Section will first introduce AI for studying, modeling, and scaling the real-world process which needs to be controlled. Thereafter the benefits of AI-based controller tuning and entirely AI-based controller are discussed. Finally, the resulting changes in the control system design procedure, raised by the introduction of AI in control system engineering, are recapitulated.

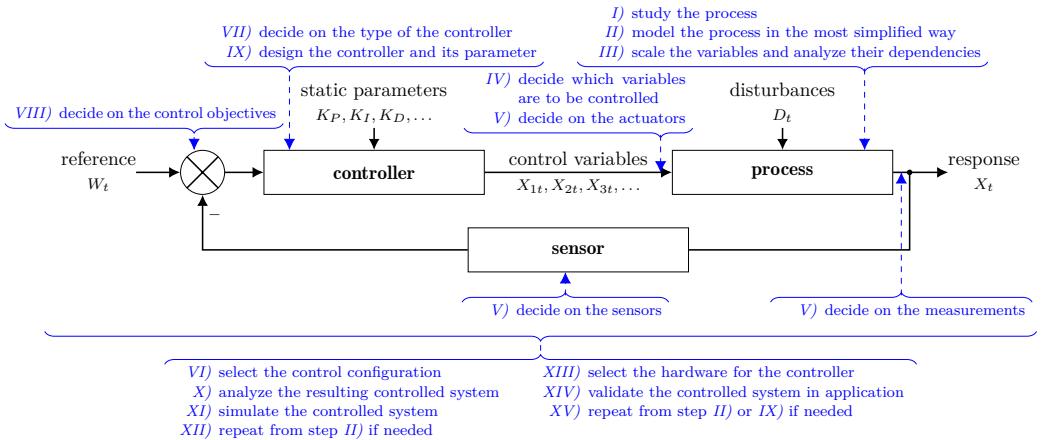


Fig. 2. The process of control system design, first 14 steps of Skogestad and Postlethwaite [26] in accordance visualized with the corresponding blocks of the CLCS block diagram.

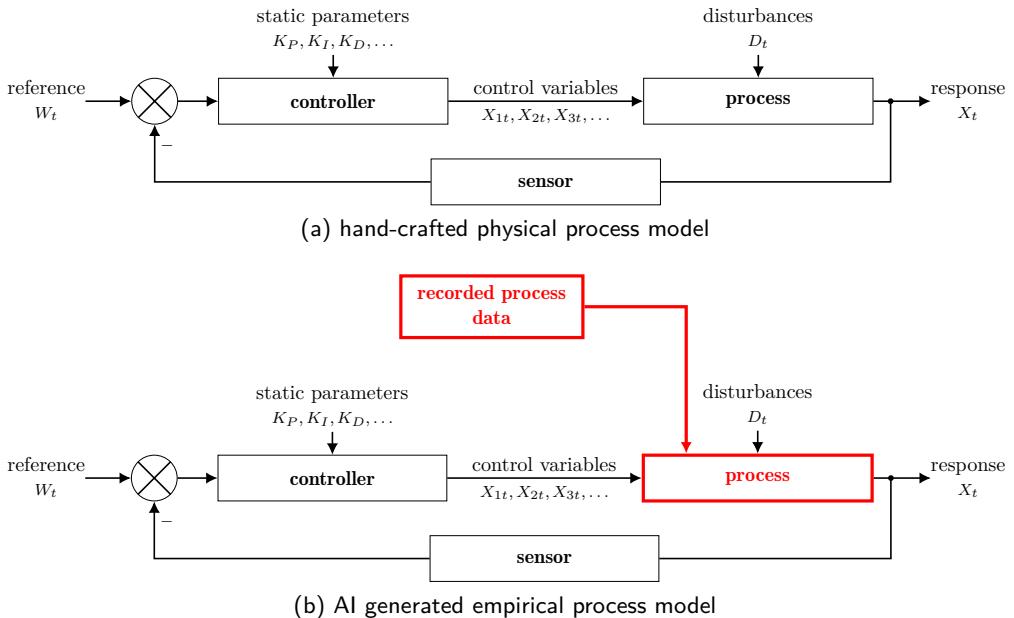


Fig. 3. CLCS block diagrams, (a) with a hand-crafted physical process model and (b) with a empirical process model, e.g. trained by an ANN on recorded data; both process models meet the needs during controller design procedure.

### 3.1 AI-based Process Modeling

By starting any control system design, the physical model of the process is studied, modeled, scaled, and evaluated. These four steps are repeated until the design model fits the real-world process, which needs to be controlled by the control system. Regarding to the

statement by Hornik et al. [11], that "standard multilayer feedforward networks are capable of approximating any measurable function to any desired degree of accuracy, in a very specific and satisfying sense", it seems to be obvious to replace the physical model with an empirical model. As illustrated in Figure 3 (b), an ANN is trained on recorded data of the process. Thus, the trained ANN as functions approximator is an empirical model of the process model. Its level of detail depends on the training data, but the needs during the controller design are met with sufficient training data. Note, in control system engineering, all data and datasets are time series; thus, sampling frequencies during data recording need to be treated during learning.

The level of detail of the AI-generated empirical process model corresponds with the amount, the variance, and the biases of the recorded training data. This means, in case the recorded data cover every assumed process variance without any biases, the realistic process model can be automated generated.

Since it is challenging to collect data from the process in border and extreme cases, a mixture of both an empirical and a physical process model is needed. Physics-informed neural networks [22] as well as neural operator [15] and their combinations [16] are promising approaches for combining these both models. Here, a physical model definition of the process covers the border and extreme cases, while the training data is considered for the general cases during the model training.

### 3.2 AI-empowered and -based Parameter Tuning

To define the optional parameter set, e.g., for PID controllers, Ziegler-Nichols tuning, Cohen-Coon tuning, Kappa-Tau tuning, heuristic tuning, and other methods are used nowadays. These methodologies lead to parameter sets, which are then commonly fine-tuned by hand, resulting in static hand-crafted control parameter, as seen in Figure 4 (a).

Next to the established tuning approaches, ANN can optimize the parameter for typical process cases based on recorded data. Illustrated in Figure 4 (b), the parameter for the controller are trained as static parameter sets so that these AI-tuned parameters can be evaluated before deploying the CLCS, ensuring that the parameter set covers the extreme cases. The advantage of AI-based, i.e., data-driven parameter tuning, is that the probability distribution of process states and the probability distribution of disturbance will be considered during training. Because, unlike the dataset for the AI-based process modeling, the data set used for parameter tuning should represent the real-world process cases. Thus the data set for parameter tuning might be strongly biased with recurrent process states.

Based on the recorded process data, an ANN can be trained for proving a dynamic set of parameter to the controller, visualized in Figure 4 (c). In this setting, the ANN might internally predict the disturbance for generating the best parameter set online. Increasing this internal representation of the current process state and the current disturbance within the ANN, the ANN needs to get sensor input directly. This sensor input can originate from the sensor, used for the feedback of the CLCS, but other additional sensors can be considered as well. With multiple sensors, the prediction of process disturbances can be optimized by training the ANN. Thus, the ANN can dynamically estimate the optimal parameter set and update in the controller. Online parameter tuning in non-convex optimization cases becomes for particular controllers, like the PID controller, mathematically an NP-hard problem [27]. Introducing AI, in the form of ANN might deal with the issue of NP-hard calculations [18] during runtime.

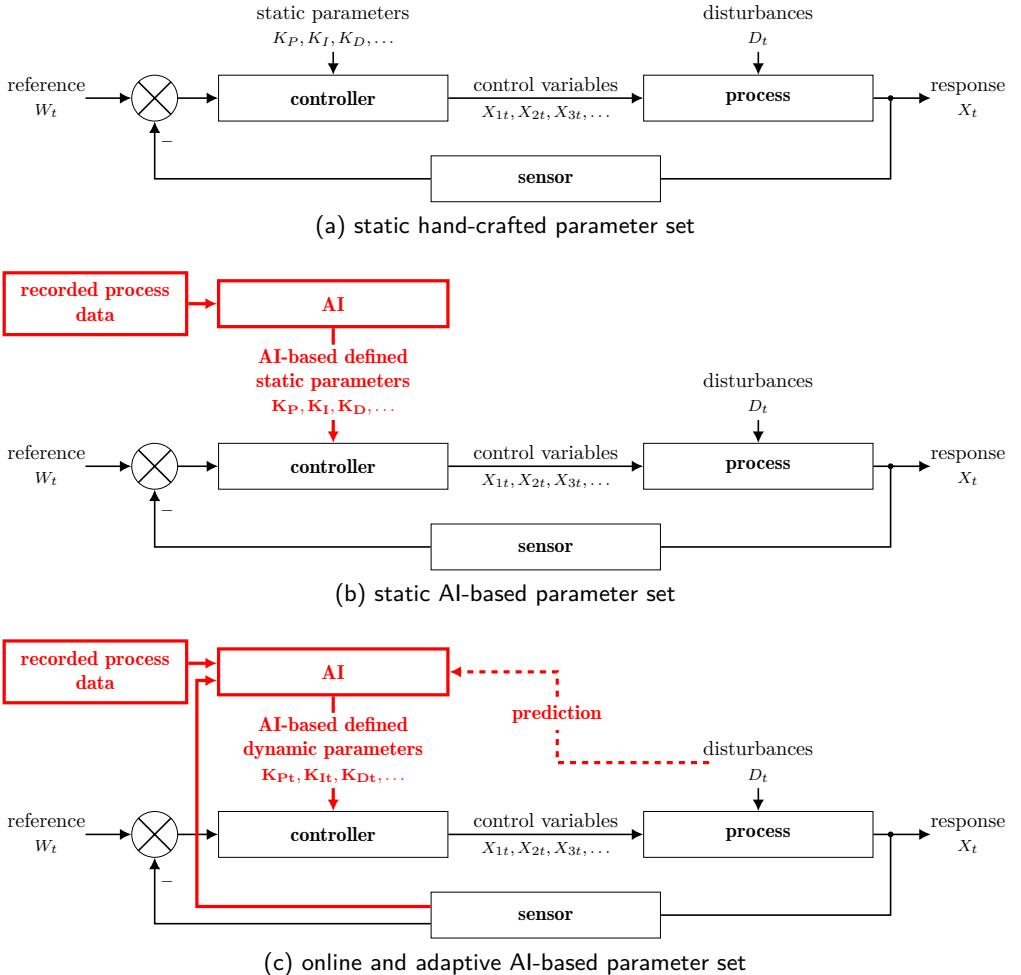
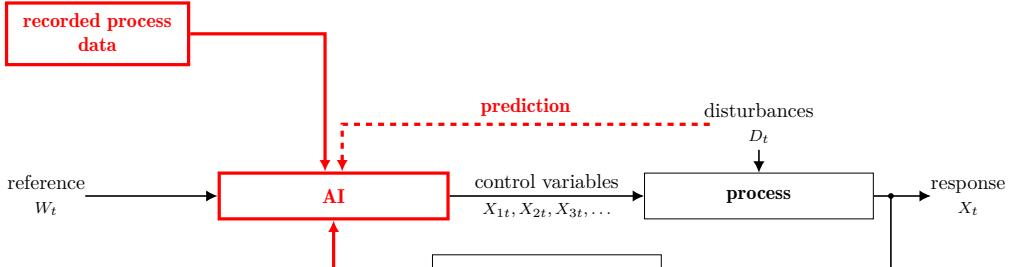


Fig. 4. Parameter tuning for CLCS, (a) fine tuned parameters by experts, (b) and (c) process data driven tuned adaptive parameters.

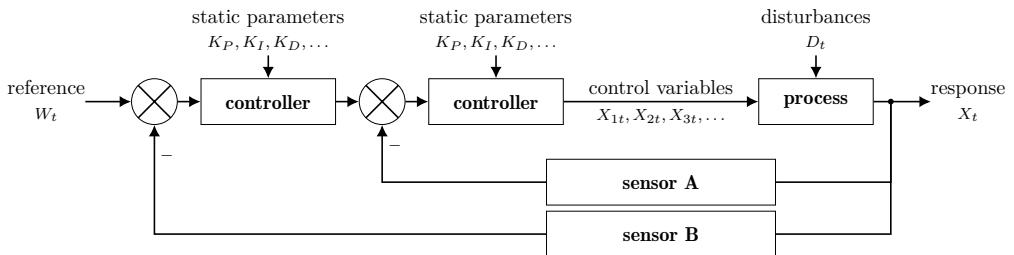
In contrast to expert rule tables for dynamical parameter tuning [17], the online estimation of controller parameters by an ANN leads to an enormous space of possible CLCS characteristics. All possible system behaviors resulting from the enormous working space of CLCS with online AI-based parameter tuning can not be simulated or tested in entirety. Consequently, AI-based dynamically online change of controller parameter sets is not recommended yet for time- and safety-critical processes.

### 3.3 AI-based Controller

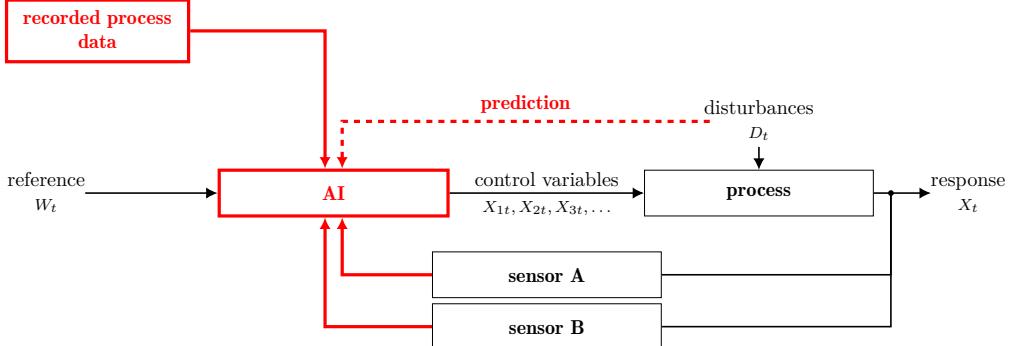
Going even a step further, the entire controller, i.e., even nested [31], and cascaded [12, 14] controller setting, can be replaced by AI-based controller. As shown in the block diagram of Figure 5 (a), a trained AI-based controller, most likely in the form of an ANN-based controller, directly sets the control variables based on the given reference value and the sensor



(a) AI-based controller



(b) cascaded setup with static parameter sets



(c) simplification due to AI-based controller

Fig. 5. AI-based controller: (a), e.g., an ANN replaces the controller design; (b) and (c) cascaded controller set up where an AI-based controller like in (c) simplifies the CLCS design.

input. Even multiple reference values, as well as sensor inputs, are directly coupled to the AI-based controller. The challenging part of CLCS design with an AI-based controller is now the training of AI. Therefore, real-world data with the typical process variance are needed and an unbiased data set, covering all variance without bias. Conserving the characteristic of both time-series data sets during learning is challenging; a possible solution is to train on both datasets simultaneously [10, 32].

In contrast, to the complex and time-expensive training of an AI-based controller, the inference times, i.e., the execution times, of trained ANN are magnitudes faster. On GPU

and particular FPGA, the inference times of deep ANN are below  $10ms$  [7, 13] and fit many processes in control. The inference times of lightweight ANNs with less than four million parameter are usual below  $3.9ms$  [13]. AI-based controllers can no longer be evaluated entirely from an evaluation and testing perspective. Even by systematically evaluating the system behavior in predefined input-values spaces, poles and extra can remain unnoticed.

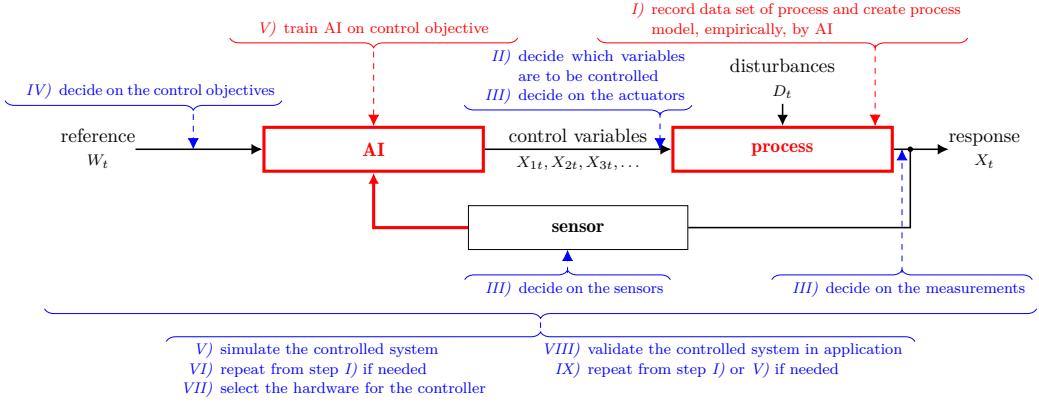


Fig. 6. The process of AI-empowered and -based control system design, 9 steps in accordance with the corresponding block of the block diagram. Blue steps identical to the steps of Skogestad and Postlethwaite [26] – red steps are altered.

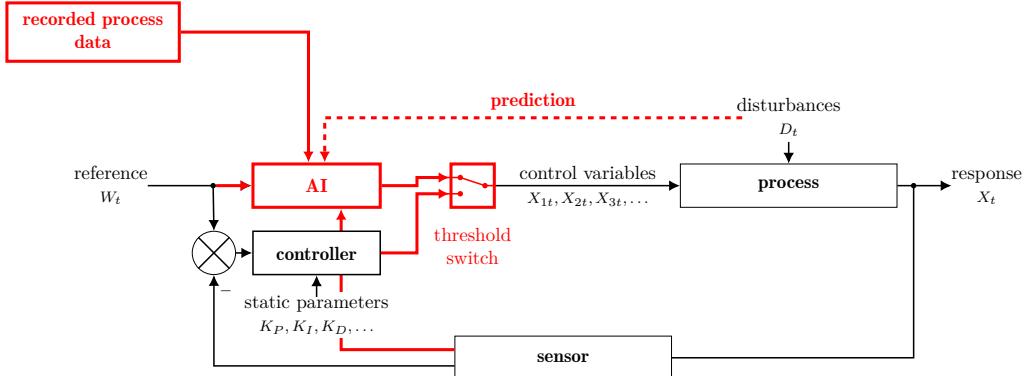
### 3.4 Procedure of the AI-empowered and -based Control System Design

Once AI can be used for and in CLCS design, the number of steps in its design will be reduced to nine steps. As shown in Figure 6, the three steps needed for describing the physical process model are condensed to a single one—recording the data set and create the empirical model automatically by the use of AI. By using an AI controller, the decision and analysis on the controller type of as well as on the control configuration become obsolete. However, similar to the tuning of the controller, the training of the AI regarding the reference values is needed.

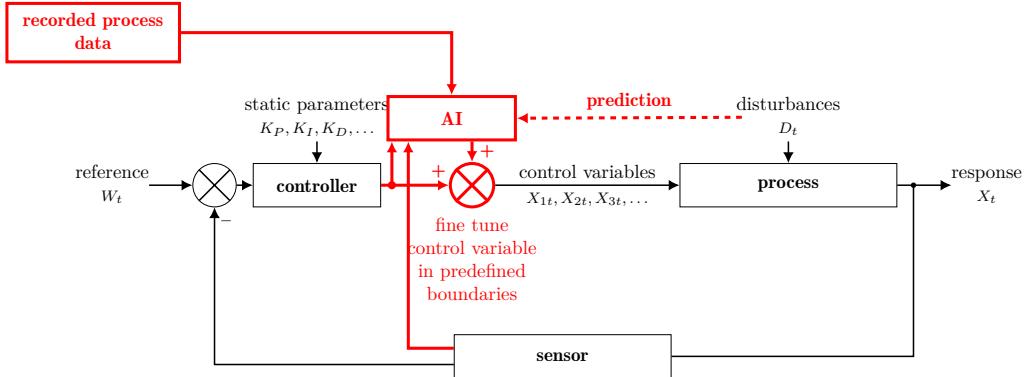
## 4 SAFETY FOR AI-EMPOWERED AND -BASED CLOSED-LOOP CONTROL SYSTEMS

AI and functional safety are nowadays mainly discussed in the context of obstacle detection for self-driving vehicles [2, 30, 33], because high accurate obstacle detection on camera, LiDar, and radar data can only be achieved by AI algorithm. The safety dilemma of AI remains unsolved until explainable AI is detailed delved.

Since explainable AI, i.e., explainable deep ANN, does not exist yet [24], AI-based controller seems no option for controlling processes. Enabling the benefits of AI-based controllers and countering the risks of CLCS impossible to evaluate and test, two possible designs are depicted in Figure 7. To guarantee functional safety, one strategy is to provide a fall-back path in combination with a safety switch, disabling the AI-based controller depending on, e.g., thresholds. The block diagram of Figure 7 (a) illustrates this design. Theoretically, implementing a fall-back path using a conventional controller seems to be trivial. Indeed, the non-trivial part is the parameterization of the switch deciding whether to pass the AI-based



(a) AI-based controller with a conventional fall-back path



(b) AI-based controller with a limited impact on the control values

Fig. 7. Controlling safety critical processes by AI-based controller without the need of explainable AI.

controller's or the fall-back controller's control variables. Further, to improve functional safety, the number of trainable weights needs to be reduced, so that works [25] on tailoring ANN needs to be considered.

Avoiding a head cut between possible unsafe AI-based controllers and a conventional ones, a second strategy is realized in Figure 7 (b). Here, a conventional controller provides the control values. The AI-based controller fine-tunes these values in predefined boundaries—usually by symmetrical boundaries by plus-minus a defined offset for each control value. Due to the predefined maximum influence of the AI-based controller on the control values, this CLCS design can be evaluated and tested to fulfill the functional safety requirements.

## 5 CONCLUSION

AI in the field of control system engineering will be exploited for creating empirical process models by ANN. AI-based process modeling from recorded process data will reduce modeling time drastically and allow the conventional without any drawback. However, recorded process data are needed to cover the variances without biases. Thus a research focus must be on

how these data can be recorded systematically, e.g., by stimulating a mechatronic actuator in a particular manner.

The AI-empowered static parameter set tuning will also be a pretty promising attempt for using the benefits of AI shortly. Nevertheless, due to its safety constraints, online AI-based changes of the controller parameters are not beneficial for most processes yet. The same lack in functional safety have entirely AI-based controllers. However, giving these kinds of controllers a defined impact on the control values of the process, as shown in Figure 7 (b), is a good starting point for studying the pros and cons of AI-based controllers within deployed CLCS.

Finally, AI in the control system domain offers the kind of momentum for new control systems design, reflecting decades of knowledge and combining these with the process information in the form of recorded data. AI will not replace the experiences needed for control system engineering but will increase the importance of systematic recording process data.

## REFERENCES

- [1] Karl J. Åström and Tore Hägglund. 2001. The future of PID control. *Control Engineering Practice* 9, 11 (2001), 1163–1175. [https://doi.org/10.1016/s0967-0661\(01\)00062-4](https://doi.org/10.1016/s0967-0661(01)00062-4)
- [2] Jyotika Athavale, Andrea Baldovin, and Michael Paulitsch. 2020. Trends and Functional Safety Certification Strategies for Advanced Railway Automation Systems. In *International Reliability Physics Symposium (IRPS)*. IEEE. <https://doi.org/10.1109/irps45951.2020.9129519>
- [3] Derek P. Atherton and Somanath Majhi. 1999. Limitations of PID controllers. In *Proceedings of the American Control Conference*. IEEE. <https://doi.org/10.1109/acc.1999.786236>
- [4] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: A fresh approach to numerical computing. *SIAM Review* 59, 1 (2017), 65–98. <https://doi.org/10.1137/141000671>
- [5] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* 113, 15 (2016), 3932–3937. <https://doi.org/10.1073/pnas.1517384113>
- [6] Cheng Siong Chin. 2017. *Computer-Aided Control Systems Design*. CRC Press. <https://doi.org/10.1201/b13697>
- [7] Javier Duarte, Song Han, Philip Harris, Sergo Jindariani, Edward Kreinar, Benjamin Kreis, Jennifer Ngadiuba, Maurizio Pierini, Ryan Rivera, Nhan Tran, and Zhenbin Wu. 2018. Fast inference of deep neural networks in FPGAs for particle physics. *Journal of Instrumentation* 13, 07 (2018), P07027–P07027. <https://doi.org/10.1088/1748-0221/13/07/p07027>
- [8] Elsevier B.V. 2022. *Scopus — Document search*. <https://www.scopus.com>
- [9] ESI Group. 2022. *Xcos*. <https://www.scilab.org/software/xcos>
- [10] Qitao Gu and Qun Dai. 2020. A novel active multi-source transfer learning algorithm for time series forecasting. *Applied Intelligence* 51, 3 (2020), 1326–1350. <https://doi.org/10.1007/s10489-020-01871-5>
- [11] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 5 (1989), 359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- [12] Jyh-Cheng Jeng and Ming-Wei Lee. 2012. Simultaneous automatic tuning of cascade control systems from closed-loop step response data. *Journal of Process Control* 22, 6 (2012), 1020–1033. <https://doi.org/10.1016/j.jprocont.2012.04.010>
- [13] Keras Team. 2022. *Keras Applications*. <https://keras.io/api/applications/>
- [14] Yongho Lee, Sunwon Park, and Moonyong Lee. 1998. PID Controller Tuning To Obtain Desired Closed Loop Responses for Cascade Control Systems. *Industrial & Engineering Chemistry Research* 37, 5 (1998), 1859–1865. <https://doi.org/10.1021/ie970769t>
- [15] Zongyi Li, Nikola Kovachki, Kamran Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. 2020. Neural Operator: Graph Kernel Network for Partial Differential Equations. (2020). arXiv:cs.LG/2003.03485v1
- [16] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamran Azizzadenesheli, and Anima Anandkumar. 2021. Physics-Informed Neural Operator for Learning Partial Differential Equations. (2021). arXiv:cs.LG/2111.03794v1

- [17] Jingwei Liu, Tianyue Li, Zheyu Zhang, and Jiaming Chen. 2020. NARX Prediction-Based Parameters Online Tuning Method of Intelligent PID System. *IEEE Access* 8 (2020), 130922–130936. <https://doi.org/10.1109/access.2020.3007848>
- [18] Furqan Memon and Cheng Shao. 2020. An Optimal Approach to Online Tuning Method for PID Type Iterative Learning Control. *International Journal of Control, Automation and Systems* 18, 8 (2020), 1926–1935. <https://doi.org/10.1007/s12555-018-0840-0>
- [19] Octave Forge Community. 2022. *Octave Forge - The 'control' package*. <https://octave.sourceforge.io/control>
- [20] Christopher Rackauckas, Alan Edelman, Keno Fischer, Mike Innes, Elliot Saba, Viral B. Shah, and Will Tebbutt. 2020. Generalized Physics-Informed Learning through Language-Wide Differentiable Programming. In *AAAI Spring Symposium: MLPS*.
- [21] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. 2021. Universal Differential Equations for Scientific Machine Learning. arXiv:cs.LG/2001.04385
- [22] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378 (2019), 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [23] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. 2020. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* 367, 6481 (2020), 1026–1030. <https://doi.org/10.1126/science.aaw4741>
- [24] Gabrielle Ras, Ning Xie, Marcel van Gerven, and Derek Doran. 2020. Explainable Deep Learning: A Field Guide for the Uninitiated. (2020). arXiv:cs.LG/2004.14545v2
- [25] Mats L. Richter, Julius Schöning, Anna Wiedenroth, and Ulf Krünnack. 2021. Should You Go Deeper? Optimizing Convolutional Neural Network Architectures without Training. In *International Conference On Machine Learning And Applications (ICMLA)*. IEEE (in press). <https://doi.org/10.1109/ICMLA52953.2021.00159>
- [26] Sigurd Skogestad and Ian Postlethwaite. 2005. *Multivariable Feedback Control* (2 ed.). John Wiley & Sons.
- [27] Oluwasegun Ayokunle Somefun, Kayode Akingbade, and Folasade Dahunsi. 2021. The dilemma of PID tuning. *Annual Reviews in Control* 52 (2021), 65–74. <https://doi.org/10.1016/j.arcontrol.2021.05.002>
- [28] Su Whan Sung and In-Beum Lee. 1996. Limitations and Countermeasures of PID Controllers. *Industrial & Engineering Chemistry Research* 35, 8 (1996), 2596–2610. <https://doi.org/10.1021/ie960090+>
- [29] The MathWorks, Inc. 2022. *The Julia Programming Language*. <https://julialang.org/>
- [30] Hans Tschurtz and Andreas Gerstinger. 2021. The Safety Dilemmas of Autonomous Driving. In *Zooming Innovation in Consumer Technologies Conference (ZINC)*. IEEE. <https://doi.org/10.1109/zinc52049.2021.9499306>
- [31] Petros G. Voulgaris. 2000. Control of nested systems. In *American Control Conference ACC*. IEEE. <https://doi.org/10.1109/acc.2000.877064>
- [32] Rui Ye and Qun Dai. 2021. Implementing transfer learning across different datasets for time series forecasting. *Pattern Recognition* 109 (2021), 107617. <https://doi.org/10.1016/j.patcog.2020.107617>
- [33] Máté Zöldy, Zsolt Szalay, and Viktor Tihanyi. 2020. Challenges In Homologation Process Of Vehicles With Artificial Intelligence. *Transport* 35, 4 (2020), 435–446. <https://doi.org/10.3846/transport.2020.12904>