electronics

Article

Safe and Trustful AI for Closed-Loop Control Systems

Julius Schöning * and Hans-Jürgen Pfisterer

Faculty of Engineering and Computer Science, Osnabrück University of Applied Sciences,
49076 Osnabrück, Germany
* Correspondence: j.schoening@hs-osnabrueck.de; Tel.: +49-541-969-7150

Abstract: In modern times, closed-loop control systems (CLCSs) play a prominent role in a wide application range, from production machinery via automated vehicles to robots. CLCSs actively manipulate the actual values of a process to match predetermined setpoints, typically in real time and with remarkable precision. However, the development, modeling, tuning, and optimization of CLCSs barely exploit the potential of artificial intelligence (AI). This paper explores novel opportunities and research directions in CLCS engineering, presenting potential designs and methodologies incorporating AI. Combining these opportunities and directions makes it evident that employing AI in developing and implementing CLCSs is indeed feasible. Integrating AI into CLCS development or AI directly within CLCSs can lead to a significant improvement in stakeholder confidence. Integrating AI in CLCSs raises the question: How can AI in CLCSs be trusted so that its promising capabilities can be used safely? One does not trust AI in CLCSs due to its unknowable nature caused by its extensive set of parameters that defy complete testing. Consequently, developers working on AI-based CLCSs must be able to rate the impact of the trainable parameters on the system accurately. By following this path, this paper highlights two key aspects as essential research directions towards safe AI-based CLCSs: (I) the identification and elimination of unproductive layers in artificial neural networks (ANNs) for reducing the number of trainable parameters without influencing the overall outcome, and (II) the utilization of the solution space of an ANN to define the safety-critical scenarios of an AI-based CLCS.

Keywords: closed-loop control systems; artificial intelligent; functional safety; trust; artificial neural networks

check for updates

Citation: Schöning, J.; Pfisterer, H.-J. Safe and Trustful AI for Closed-Loop Control Systems. *Electronics* **2023**, *12*, 3489. https://doi.org/10.3390/electronics12163489

Academic Editor: Padma Iyenghar

Received: 5 July 2023
Revised: 10 August 2023
Accepted: 15 August 2023
Published: 17 August 2023

1. Introduction

In an ideal scenario, a closed-loop control system (CLCS) controller of processes or systems would precisely set the control variables to match the desired reference value perfectly. The intended process response would consistently align with the reference value, irrespective of the intensity of disturbances affecting the process. However, real-world processes and systems, such as mechatronic actuators in vehicles, biochemical processes in reactors, and building temperature regulation systems, inevitably exhibit deviations such as overshoots, settling times, and oscillations during control. Therefore, technology that brings real-world controllers closer to the theoretical ideal controllers is highly valued. Despite advancements in computational power, the general procedure for designing control systems [1,2] has remained essentially unchanged over the past four decades.

The introduction of artificial intelligence (AI) in the realm of closed-loop and feedback control systems, hereafter referred to as CLCSs, holds the potential to enhance the overall performance of CLCSs and the behavior of controllers, particularly for complex and nonlinear processes. Furthermore, AI can forecast potential disturbances based on, e.g., collected data and external multivariate inputs like weather forecasts, even before these disturbances impact the process. It is conceivable that AI could entirely replace the traditional controller block and directly interpret sensor values as process feedback. Consequently, using AI will reshape the existing block diagrams of CLCSs and the controller design process for supporting the integration of AI and external multivariate inputs within control system engineering [3].

Through exploring novel opportunities and research directions toward safe AI-based CLCSs, this paper aims to identify the research gap in this field. Therefore, by extending the conference paper on AI-based CLCSs [3], the following sections will delve into various aspects of CLCSs and the integration of physics-informed AI. Section 2 provides a comprehensive summary of the pipeline of applying AI, the current state of the art in CLCSs, and physics-informed AI. In Section 3, block diagrams empowered by AI and their application in CLCSs will be presented, along with detailed explanations of new blocks and design patterns. Section 4 shows the use of minimal human–machine interactions (HMI) for having AI-in-the-loop. Safety is a paramount concern for CLCSs, particularly in specific processes, and Section 5 explores how AI-based CLCSs can be integrated with functional safety measures. Lastly, in Section 6, this paper concludes by discussing the advantages and disadvantages of AI-empowered and AI-based CLCS designs while highlighting potential research directions for utilizing AI within CLCSs. For the sake of simplicity, the term "process" is used interchangeably in this paper to refer to both the controlled process and the controlled systems. Similarly, the term "system" represents the entirety of the open- or closed-loop system hereafter.

2. State of the Art

While performing an intensive review of published work and source code, we noticed that AI, particularly artificial neural networks (ANNs), has not yet made significant advancements in the field of control systems, encompassing both open-loop and closed-loop systems. The quantitative assessment of keywords from the Elsevier Scopus bibliographic database supports this qualitative impression. By plotting the curve gradients between the publication count of the keyword *Neural Network* and the combined keywords *Closed-Loop Control Systems* and *Neural Network*, the gradients for the keywords *Neural Network* and *Closed-Loop Control Systems* show exponential growth but the gradient for the combined keywords *Closed-Loop Control Systems* and *Neural Network* only linear growth. In absolute numbers from the year 2021, the research gap becomes visible as well, i.e., the single keywords *Neural Network* and *Closed-Loop Control Systems* have been associated with more than 6000 publications, but the combined keywords only with 83 publications [3]. Several reasons can account for the limited integration of control system engineering and AI thus far:

- The complexity of the phase space: The phase space described by the differential equations governing control systems is inherently complex. This complexity poses challenges for applying AI techniques, including ANN, which require substantial computational resources and training data to accurately represent and model such complex systems.
- The unpredictability of disturbances: Control systems must contend with unpredictable disturbances affecting the controlled process. These disturbances introduce uncertainties that can make it difficult for AI methods, including ANN-based approaches, to adapt and respond effectively to real-time changing conditions.
- The tradition in control system engineering: The field of control system engineering has traditionally relied on comprehensive physical models to describe processes in their entirety. This emphasis on physical models, rooted in established practices, may have hindered the integration of AI, including ANN, which operate based on data-driven approaches that may deviate from traditional modeling paradigms.

Addressing these challenges will require concerted research and development efforts to bridge the gap between control system engineering and AI. By exploring innovative approaches and leveraging advancements in AI, it is possible to overcome these obstacles and unlock the potential for AI-driven control systems in the future.

2.1. Pipeline of Applying AI

The AI field has witnessed a trend of increasingly deeper architectures, e.g., this can be observed in Figure 1, where the number of parameters in ANN has exponentially increased from InceptionV2 [4] published in 2015 to BASIC-L [5] in 2023, surpassing an 18,650.0% growth of the trainable parameter. However, accuracy increased by 21.66% from 74.8% of InceptionV2 to 91.0% of BASIC-L. The continuous deepening of ANN necessitates more hardware resources, even for inference alone, leading to long execution times. Moreover, deeper ANN architectures amplify hardware requirements and demand larger datasets for training. When considering AI-based CLCS in embedded devices, the question arises: Should one continue to pursue deeper architectures [6,7]?

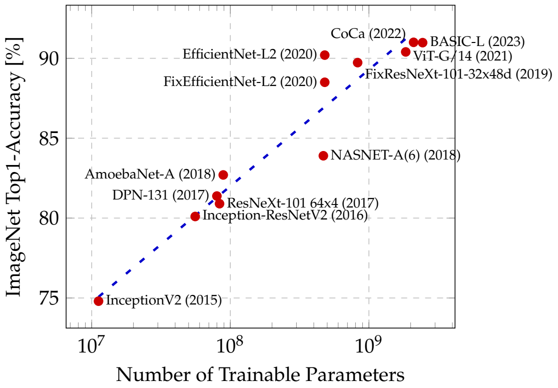


Figure 1. In AI applications where functional safety and trustworthiness are not critical requirements, a noticeable trend emerges when examining the ANN architectures developed: predictive performance improves linearly, while the number of trainable parameters and, thus, computational complexity increase exponentially [8].

The development of AI can be theoretically described as a six-step process. Figure 2 depicts this process, which begins with the availability of data and an application idea. Whether the idea or the data comes first can be seen as a chicken-and-egg dilemma. Typically, ideas reshape the data collection process, and conversely, the data can inspire new ideas, creating an iterative cycle of refinement and improvement. With the introduction of intuitive HMI and user interfaces (UI) for AI-based applications, AI-based CLCS and its development will become more applicable.

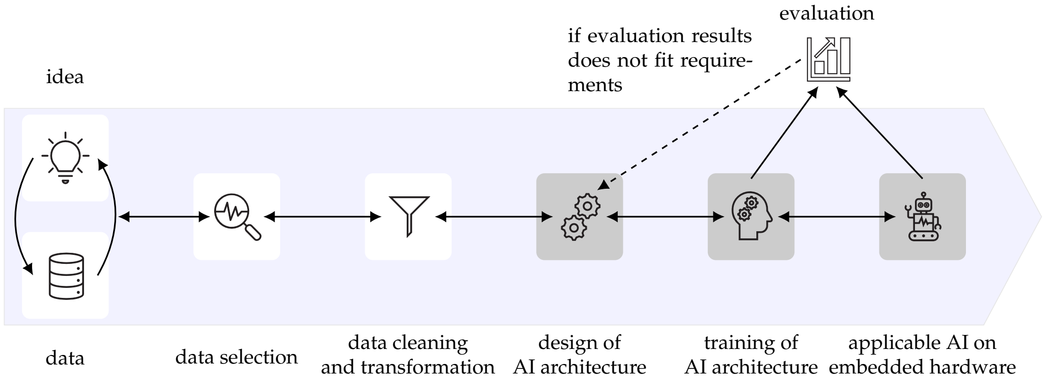

evaluation
if evaluation results does not fit requirements
idea
data
data selection
data cleaning and transformation
design of AI architecture
training of AI architecture
applicable AI on embedded hardware

Figure 2. The AI application pipeline involves six steps, steps where interactive HMI commonly used is depicted on a white background, while interactive HMI is not yet used on steps with a gray background [8].

Reflecting the needs of CLCSs, interactive HMI for two steps are essential for CLCS design. Gaining the trustfulness of the AI-based CLCS, the design of the AI architecture should highlight each layer’s impact on the overall outcome, helping to create tiny ANN architectures. For the safety assessment, the evaluation step is vital, and by utilizing an n-dimensional solution space, the operation range of the AI-based CLCS enables border case testing of the AI-based CLCS.

2.2. Closed Loop Control Systems

The proportional–integral–derivative (PID) controller is widely used in CLCS. Its popularity stems from the fact that engineers have a good understanding of how to adjust its three parameters [9]—proportional (P), integral (I), and derivative (D)—based on past, present, and future control errors [10], to achieve the desired CLCS behavior. However, PID controllers face limitations regarding nonlinear processes and processes with short dead times [9]. Depending on their parameters, PID controllers can also exhibit tendencies to overshoot, undershoot, or exhibit slow response in reaching the setpoint [11]. In response to these challenges, strategies for tuning the PID controller parameters [11] have been optimized, and more complex designs, such as PID-P, PI-PD, and PI-D controllers, have been proposed as alternatives [9].

Controlling nonlinear processes with nonperiodic uncertainties remains a difficult task [10]. Moreover, conventional CLCS design approaches are inadequate for managing processes with multivariable responses and references.

2.3. AI-Based Closed Loop Control Systems

Recently, the absence of a connection between artificial intelligence (AI) and control system engineering has been acknowledged, leading to the emergence of first scientific papers addressing this gap. Rackauckas et al. [12] presented an innovative approach utilizing ANN to approximate and even discover missing terms in differential equations. Their methodology, known as universal differential equations discovery, enables the approximation of specific process equations based on recorded data.

To combine AI and CLCS, differentiable programming languages like Julia [13] are crucial for solving differential equations and training ANN within the same software code. The advantages of Julia as a programming language have been demonstrated by Rackauckas et al. [14] and its suitability for this purpose. Another related concept is that of physics-informed neural networks [15], which shares the same foundational idea but may employ different programming languages to integrate differential equations and ANN.

Two notable review papers deserve mention, focusing on nonlinear [16] and dynamic [17] differential equations in combination with ANN, respectively. Despite these advancements, the gap between control engineering and AI still persists in engineering applications.

2.4. Procedure of the Control System Design

Since the start of the new millennium, there has been a notable rise in the availability of computer-aided software solutions for applied control design. Prominent examples include Simulink (MATLAB) [18], Xcos (Scilab) [19], and control (Octave) [20], which embody the current best practices in controller design procedures. Figure 3 illustrates the typical progression encompassing 14 common steps involved in transitioning from the initial process requiring control to the final design of the control system [1]. Additionally, based on personal experience, an additional 15th step is incorporated in Figure 3.

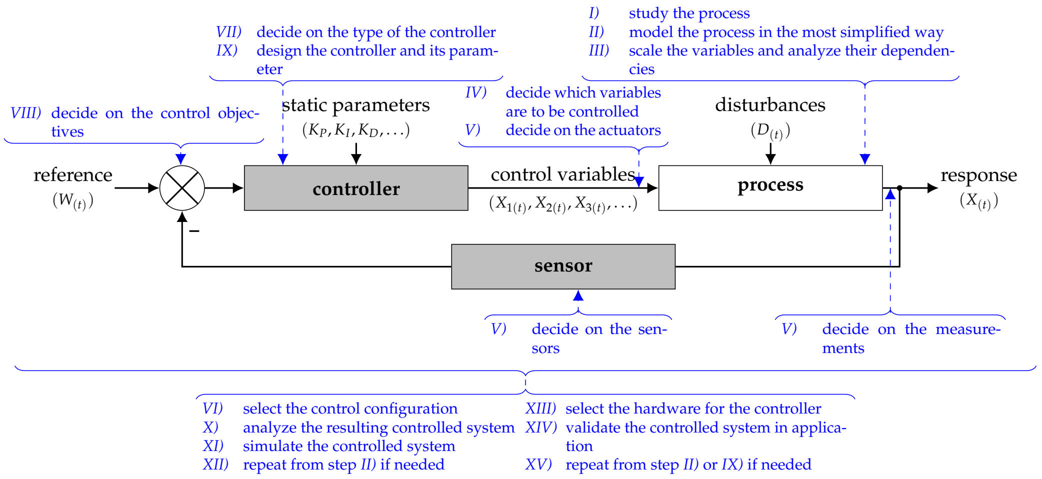

I) study the process
II) model the process in the most simplified way
III) scale the variables and analyze their dependencies
VII) decide on the type of the controller
IX) design the controller and its parameter
VIII) decide on the control objectives
static parameters
$(K_P, K_I, K_D, \ldots)$
IV) decide which variables are to be controlled
V) decide on the actuators
disturbances
$(D_{(t)})$
reference
$(W_{(t)})$
controller
control variables
$(X_{1(t)}, X_{2(t)}, X_{3(t)}, \ldots)$
process
response
$(X_{(t)})$
−
sensor
V) decide on the sensors
V) decide on the measurements
VI) select the control configuration
X) analyze the resulting controlled system
XI) simulate the controlled system
XII) repeat from step II) if needed
XIII) select the hardware for the controller
XIV) validate the controlled system in application
XV) repeat from step II) or IX) if needed

Figure 3. The process of control system design, first 14 steps of Skogestad and Postlethwaite [1] in accordance visualized with the corresponding blocks of the CLCS block diagram—blocks with gray background are crucial for safety of the CLCS.

3. AI Empowerment for Closed-Loop Control Systems

When examining the control system design procedure and the currently used CLCS block diagrams, AI can be utilized in various design steps and blocks. This section will explore the application of AI in studying, modeling, and scaling real-world processes that require control. Subsequently, this section will delve into the advantages of AI-based controller tuning and the possibility of entirely AI-based controllers. Lastly, it will summarize the alterations in the control system design procedure that arise from the integration of AI in control system engineering.

3.1. Functional Safety and Trustfulness

Before equipping CLCSs with AI, the question arises as to which blocks of a CLCS are mainly responsible for a system's functional safety and trustworthiness. When considering a possible use case of a CLCS in an automated vehicle, the functional safety of a deployed CLCS is evaluated using ISO 26262 [21]. ISO 26262 defines the analysis of systems with respect to an automotive safety integrity level as a standard. According to ISO 26262, the item definition of a CLCS states that the controller must set the controlled variable precisely to a desired setpoint. Through the following hazard analysis and risk assessment, it is clear that the control or the AI and the sensor block are critical to the safety of a CLCS, as shown in Figures 3 and 7.

Assessing trust in automation is a difficult task [22–24]. Trust has been shown to change over time with accumulated experience [25,26]. Consequently, all stakeholders of CLCSs, i.e., developers, testers, assessors, end users, etc., need to experience the safeness of AI-based CLCSs to gain trust over time.

3.2. AI-Based Process Modeling

When embarking on a control system design, it is customary to study, model, scale, and evaluate the physical characteristics of the process. These iterative steps continue until the design model accurately represents the real-world process to be controlled by the control system. Given the assertion by Hornik et al. [27] 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 becomes apparent that the physical model can be replaced with an empirical model. In Figure 4b, an AI is trained on recorded process data, thus serving as an empirical approximation of the process model. The level of detail of this AI-generated empirical model depends on the training data, but it is sufficient for the purposes of controller design. It is important to note that in control system engineering, all data and datasets are typically time series, necessitating careful consideration of sampling frequencies during the data recording and learning processes.

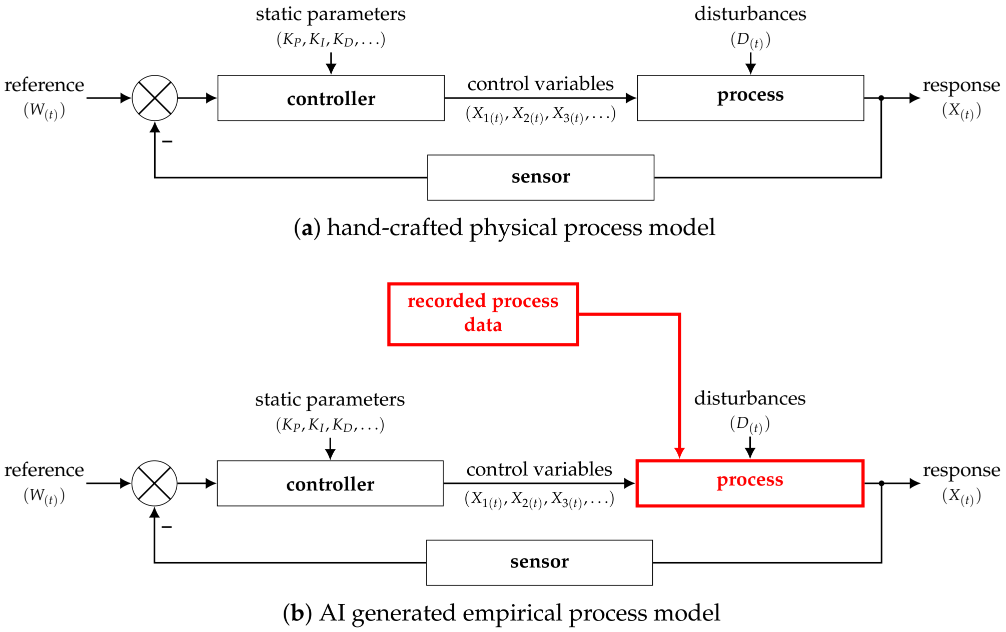


Figure 4. CLCS block diagrams: (**a**) with a typical hand-crafted physical process model and (**b**) with an empirical process model that, e.g., is generated by an AI on recorded and synthetic data; both process models meet the needs during the controller design procedure [3].

The level of detail in the AI-generated empirical process model corresponds to the quantity, variance, and biases present in the recorded training data. A realistic process model can be automatically generated if the recorded data encompass the entire range of assumed process variances without any biases.

However, since gathering data that adequately capture border and extreme cases of the process can be challenging, a combination of empirical and physical process models is often necessary. Promising approaches for integrating these models include physics-informed neural networks [15], neural operators [28], and their combinations [29]. In this context, the physical model defines the process behavior in border and extreme cases, while the training data are used to capture general cases during the model training phase. Furthermore, approximated physical process models can be used to generate a synthetic dataset, which, combined with recorded data, can be used as a hybrid dataset for training the AI-based process model.

3.3. AI-Empowered and -Based Parameter Tuning

To determine the optimal parameter set, such as for PID controllers, various methods are employed today, including Ziegler–Nichols tuning, Cohen–Coon tuning, Kappa–Tau tuning, heuristic tuning, and others. These methodologies result in parameter sets that are commonly fine-tuned manually, leading to static hand-crafted control parameters as depicted in Figure 5a.

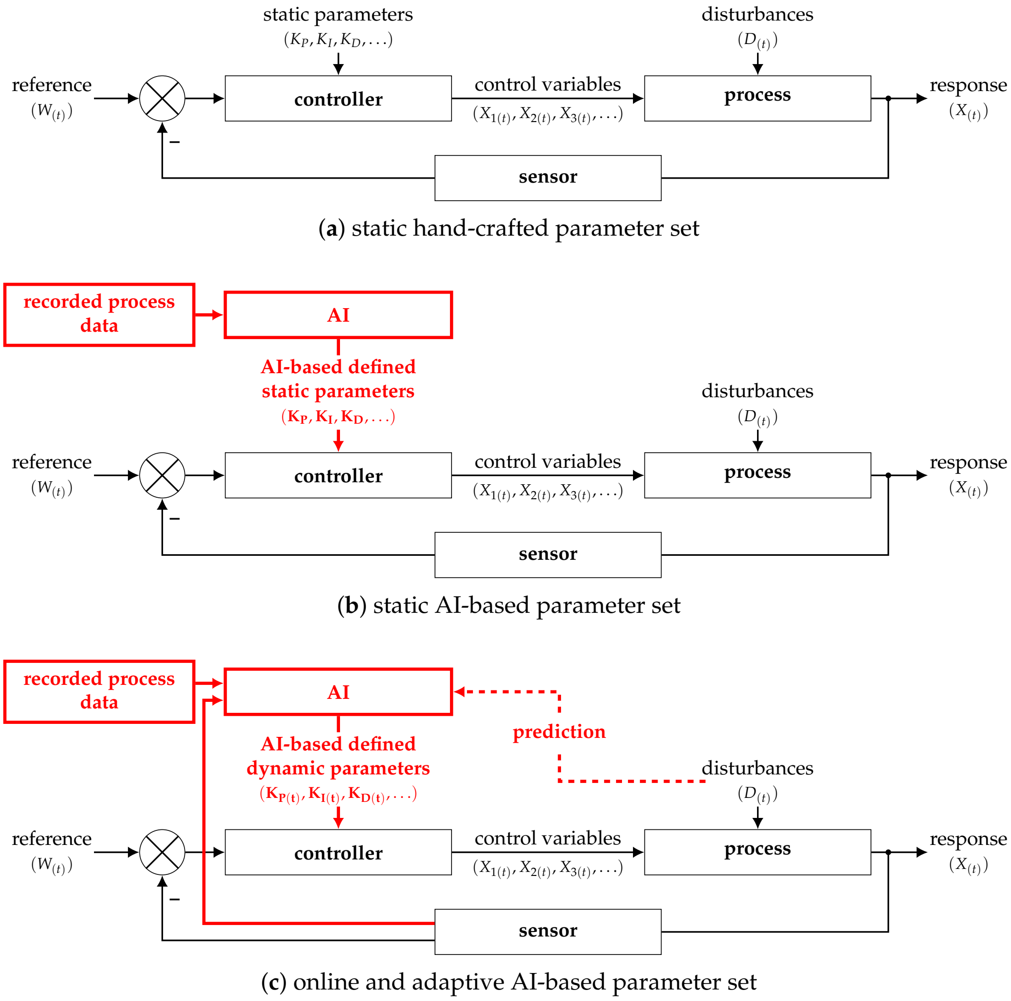

static parameters
$(K_P, K_I, K_D, \ldots)$
disturbances
$(D_{(t)})$
reference
$(W_{(t)})$
controller
control variables
$(X_{1(t)}, X_{2(t)}, X_{3(t)}, \ldots)$
process
response
$(X_{(t)})$
−
sensor
(**a**) static hand-crafted parameter set
recorded process data
AI
AI-based defined static parameters
$(K_P, K_I, K_D, \ldots)$
(**b**) static AI-based parameter set
prediction
AI-based defined dynamic parameters
$(K_{P(t)}, K_{I(t)}, K_{D(t)}, \ldots)$
(**c**) online and adaptive AI-based parameter set

Figure 5. Three different parameter tuning options for CLCSs: (**a**) finely tuned parameters by experts as common, (**b**,**c**) AI-based data-driven tuning of adaptive control parameters [3].

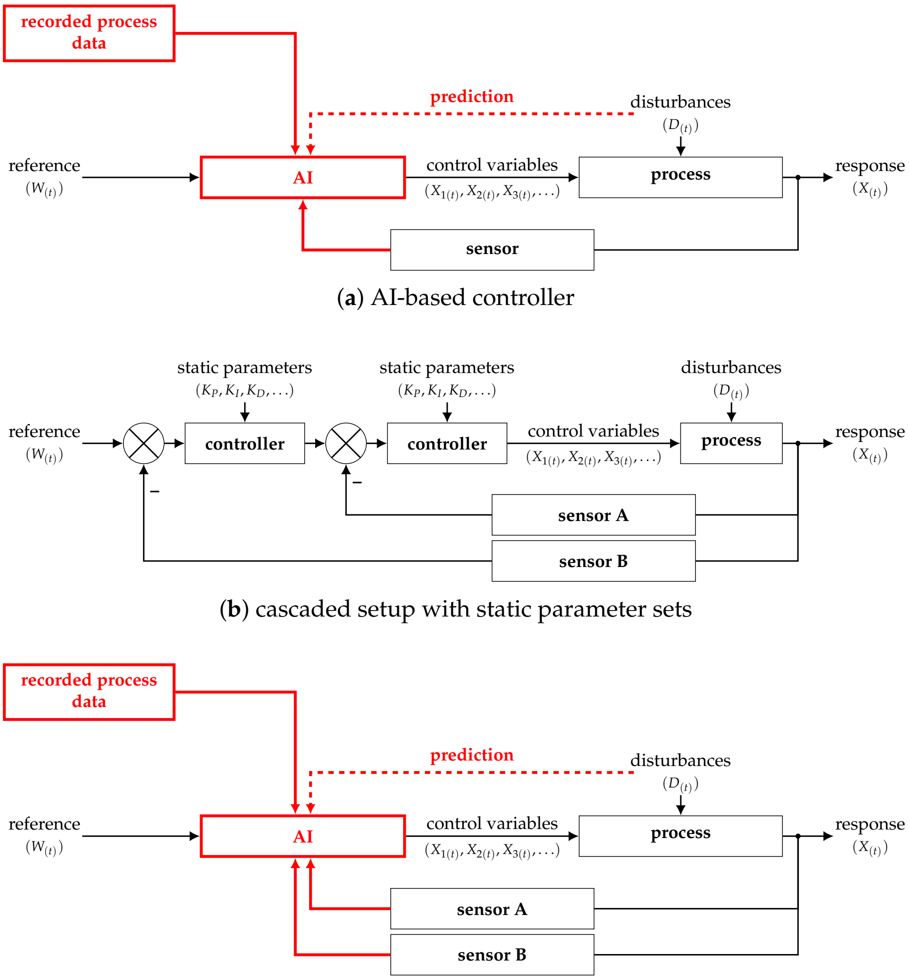
In addition to traditional tuning approaches, ANN can optimize controller parameters based on recorded data for typical process scenarios. Figure 5b shows that the controller parameters are trained as static parameter sets using AI. Using AI for parameter tuning only preserves the standard safety evaluation before deploying the CLCS, ensuring the coverage of extreme cases. The advantage of AI-based parameter tuning, driven by data, is that it considers the probability distribution of process states and disturbances during training. Unlike the dataset used for AI-based process modeling, the dataset for parameter tuning should accurately represent real-world process cases. Consequently, the data used for parameter tuning may exhibit strong biases with recurrent process states.

By leveraging recorded process data, an ANN can be trained to provide a dynamic set of parameters for the controller, as illustrated in Figure 5c. In this configuration, the ANN may internally predict disturbances to generate the best parameter set in real time, i.e., during controlling the actual process. Direct sensor inputs are required to enhance the ANN's representation of the current process state and disturbance. This sensor input can originate from the CLCS feedback sensor, but additional sensors can also be incorporated. With multiple sensors, the ANN can optimize the prediction of process disturbances through training. Thus, the ANN can dynamically estimate the optimal parameter set and update the controller. It is worth noting that online parameter tuning in nonconvex optimization cases, such as for PID controllers, poses a mathematically NP-hard problem [30]. Introducing AI in the form of ANNs may help address the computational challenges associated with NP-hard calculations during runtime [31].

Unlike expert rule tables for dynamic parameter tuning [32], the online estimation of controller parameters through an ANN introduces a vast range of potential characteristics for the CLCS. Due to the sheer magnitude of the CLCS working space with online AI-based parameter tuning, it is practically impossible to simulate and test all possible system behaviors comprehensively. As a result, the dynamic online change of controller parameter sets based on AI is currently not recommended for time- and safety-critical processes, where reliability and safety are of utmost importance.

3.4. AI-Based Controller

Taking a step further, it is possible to replace the entire controller, including nested [33] and cascaded [34,35] controller settings, with an AI-based controller. In the block diagram depicted in Figure 6a, an AI-based controller, typically in the form of an ANN-based controller, is trained to directly manipulate the control variables based on the given reference value and sensor input. This AI-based controller can handle multiple reference values and sensor inputs simultaneously. The main challenge in designing a CLCS with an AI-based controller lies in training the AI itself. It requires real-world data encompassing typical process variances in an unbiased manner, covering the entire range of variances. Balancing the characteristics of both time-series datasets during training poses a significant challenge, and one possible solution is to train on both datasets simultaneously [36,37].



(**a**) AI-based controller

(**b**) cascaded setup with static parameter sets

(**c**) simplification due to AI-based controller

Figure 6. AI-based controller: (**a**) e.g., an ANN replaces the controller design; (**b**,**c**) cascaded controller setup where an AI-based controller like in (**c**) simplifies the CLCS design [3].

In contrast to an AI-based controller's complex and time-consuming training process, the inference times, i.e., the execution times, of trained ANNs are significantly faster. Deep ANNs can achieve inference times below 10 ms on graphics processing units (GPUs), tensor processing units (TPUs) and field programmable gate array (FPGA) [38,39], making them suitable for controlling many processes. Furthermore, lightweight ANNs with fewer than four million parameters typically have inference times below 3.9 ms [39]. However, evaluating AI-based controllers solely based on testing and evaluation becomes challenging. Even with systematic evaluations of system behavior in predefined input-value spaces, certain aspects such as poles and other anomalies may go unnoticed.

3.5. Procedure of the AI-Empowered and -Based Control System Design

Integrating AI into CLCS design reduces the process to a nine-step process. As depicted in Figure 7, the three steps required to describe the physical process model are condensed into a single step, which involves recording the dataset and automatically creating the empirical model using AI techniques. By employing an AI controller, the need for decision making and analysis regarding the controller type and control configuration becomes unnecessary. However, similar to tuning the controller, training the AI model with respect to the reference values is still required.

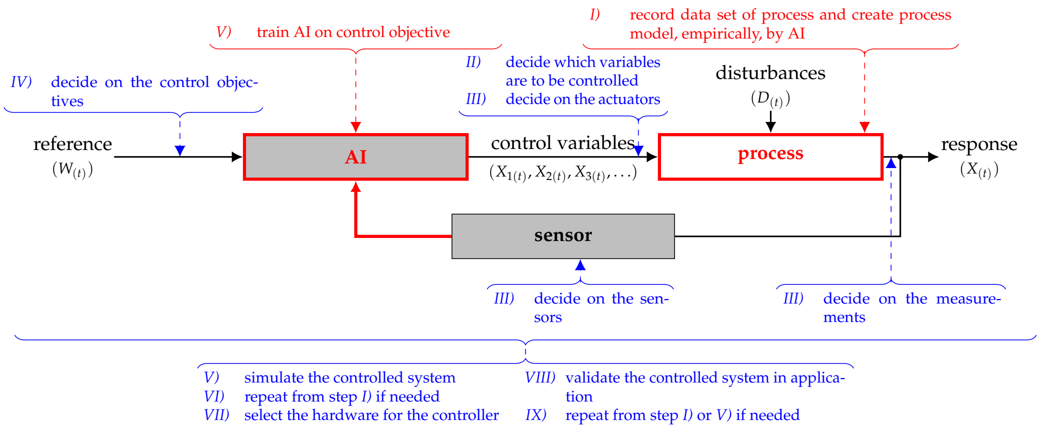

I) record data set of process and create process model, empirically, by AI
V) train AI on control objective
IV) decide on the control objectives
II) decide which variables are to be controlled
III) decide on the actuators
disturbances
$(D_{(t)})$
reference
$(W_{(t)})$
AI
control variables
$(X_{1(t)}, X_{2(t)}, X_{3(t)}, \ldots)$
process
response
$(X_{(t)})$
sensor
III) decide on the sensors
III) decide on the measurements
V) simulate the controlled system
VI) repeat from step I) if needed
VII) select the hardware for the controller
VIII) validate the controlled system in application
IX) repeat from step I) or V) if needed

Figure 7. The process of AI-empowered and -based control system design, 9 steps in accordance with the corresponding block of the block diagram. Blue steps are identical to the steps of Skogestad and Postlethwaite [1]—red steps are altered; blocks with gray background are crucial for safety of the CLCS.

4. AI-in-the-Loop

An often overlooked solution is the implementation of a minimal HMI to ensure safe and trustful task execution when the AI algorithm encounters challenges by, e.g., unknown disturbance during inference. By merging the concept of "user-in-the-loop" with AI, the AI takes over the main loop of operation while the user's role transforms into that of a domain expert and supervisor for corner cases. This concept, illustrated in Figure 8, revolves around the idea that the AI performs the majority of the workload, with the user providing guidance and support in addressing safety, security, and task-related issues that the AI architectures may struggle with. By introducing HMI at this stage of the AI application pipeline, embedded AI-based applications like AI-based CLCS [3], interactive 3D reconstruction [40], and AI-empowered crop rotation [41] can be brought to market sooner than fully automated AI-based systems.

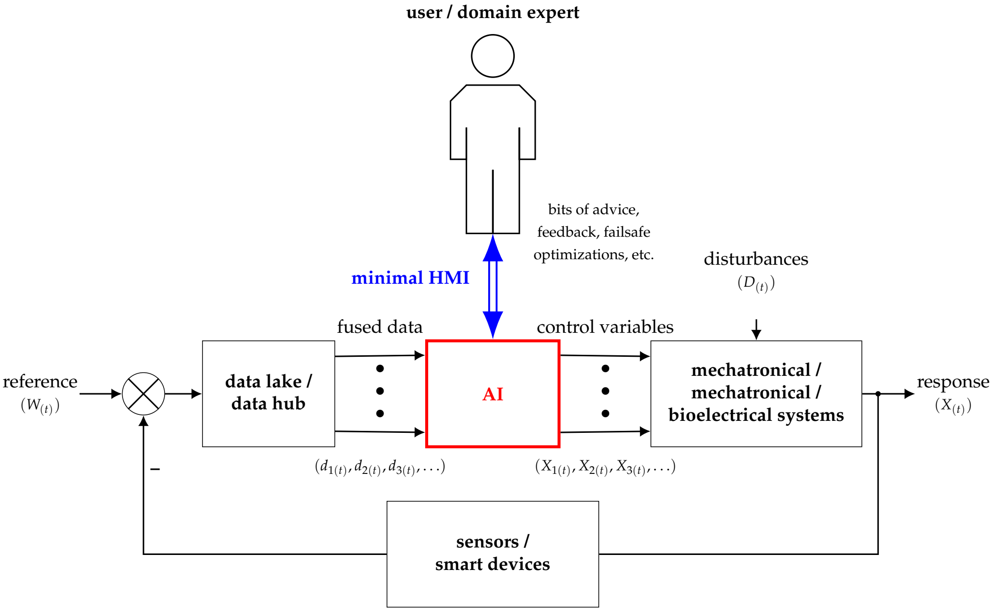

user / domain expert
bits of advice, feedback, failsafe optimizations, etc.
minimal HMI
disturbances
$(D_{(t)})$
fused data
control variables
reference
$(W_{(t)})$
data lake / data hub
AI
mechatronical / mechatronical / bioelectrical systems
response
$(X_{(t)})$
$(d_{1(t)}, d_{2(t)}, d_{3(t)}, \ldots)$
$(X_{1(t)}, X_{2(t)}, X_{3(t)}, \ldots)$
–
sensors / smart devices

Figure 8. The AI-in-the-loop concept enables the reliable, secure, and trustworthy use of AI. AI serves as the working horse, i.e., effectively handling tasks and, if needed, incorporating user expertise with minimal UI [8].

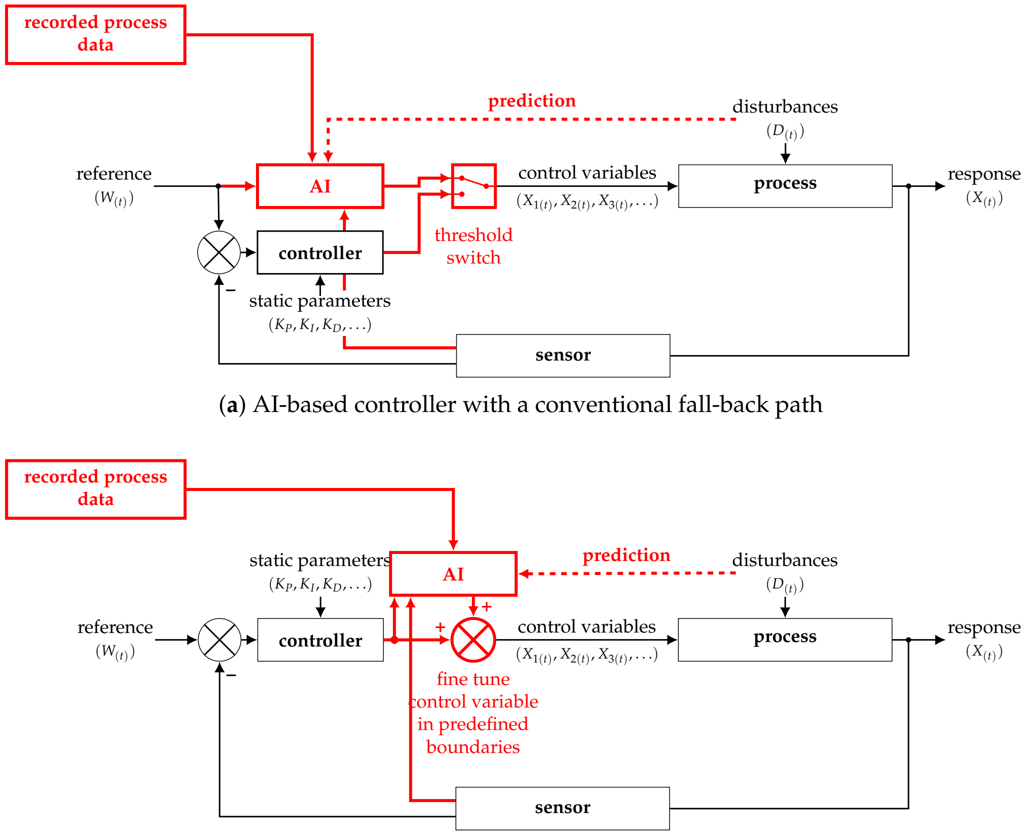
The AI-in-the-loop approach combines the computational capabilities of AI with the conceptual strengths of human interaction by enabling HMI and UI during AI inference. The AI remains the primary "working horse", using its computational power to solve the task at hand. However, if the AI encounters uncertainty in its results or identifies a safety- or security-critical situation, it seeks input from the user through an appropriate HMI with an interactive UI. Developing intuitive HMI and UI is crucial, as it requires translating the AI's trained high-dimensional task representation into a human-understandable problem description. Some UI methods already exist for visualizing and explaining the AI's current output [42–44], but further research is needed to align them with the specific requirements and preferences of users within the AI-in-the-loop concept.

By combining interactive AI architecture design and streamlining the structure of lightweight ANNs, the development of a minimal HMI to support AI-in-the-loop functionality will pave the way for the safe and reliable use of AI in the near future. Rather than aiming for fully AI generated CLCSs where the user's workload is eliminated entirely, the logical progression is expected to involve human–AI cooperation within AI-in-the-loop systems. It is important to note that AI-based CLCSs currently have unknown blind spots until the field of explainable AI is further explored and developed.

5. Functional Safety for AI-Empowered and -Based Closed-Loop Control Systems

Currently, AI and functional safety are primarily being discussed in relation to obstacle detection in self-driving vehicles [45–47]. AI algorithms have proven to be highly effective in accurately detecting obstacles using camera, LiDAR, and radar data. However, the safety concerns surrounding AI still persist, and the challenge of ensuring explainable AI remains unresolved. Further exploration and research are required to address these issues and enhance the safety aspects of AI in various domains.

Since explainable AI, specifically explainable deep ANN, has not yet been achieved [48], the use of AI-based controllers may not be a viable option for controlling processes. Two possible designs are proposed to address the challenges of evaluating and testing complex AI-based control systems and ensure functional safety, as depicted in Figure 9.



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

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

Figure 9. Two designs of controlling safety-critical processes by AI-based controller without the need for explainable AI [3].

One approach to guarantee functional safety involves incorporating a fallback path combined with a safety switch, as shown in Figure 9a. This design involves implementing a conventional controller as a backup, and a safety switch is employed to disable the AI-based controller based on predefined thresholds or conditions. While the implementation of a fallback path using a conventional controller may seem straightforward in theory, the parameterization of the switch to determine whether the control variables should be passed through the AI-based controller or the fallback controller is nontrivial. It is important to carefully consider the design of the AI-based controller to minimize the number of trainable weights, reducing complexity and improving functional safety. Tailoring the ANN based on specific needs should also be considered in this design [6,7].

An alternative strategy, as depicted in Figure 9b, involves using a conventional controller to provide the control values while the AI-based controller fine-tunes these values within predefined boundaries. Typically, symmetrical boundaries are used, allowing the AI-based controller to adjust the control values within a defined offset. This design ensures that the influence of the AI-based controller on the control values remains within predefined limits, facilitating evaluation and testing to meet functional safety requirements.

These strategies aim to address the challenges associated with AI-based controllers and ensure functional safety by either incorporating fallback paths and safety switches or constraining the AI-based controller's influence on the control values within predefined boundaries.

6. Research Directions and Conclusions

The field of control system engineering is poised to leverage AI to create empirical process models using ANN. By employing AI-based process modeling techniques with recorded and synthetically generated process data, the time required for modeling can be drastically reduced without any drawbacks to conventional methods. By exploring novel opportunities in the abovementioned sections, it becomes clear that the following fields need to be researched to unleash the potential of AI-based CLCSs.

6.1. Research Direction: Tiny AI

Since the complexity of the phase space described by the differential equations governing control systems is inherently complex, applying AI to CLCSs will probably lead to ANNs with a high number of trainable parameters. The high number of parameters will lead to ANNs that classical function safety evaluations cannot cover due to the size of the solution space spanned by the used ANNs. Thus, research should focus on tiny ANNs, i.e., with few trainable parameters. As one possible option to create tiny ANNs, the elimination of unproductive layers of an ANN is already feasible [6,7].

6.2. Research Direction: Solution Space of ANN

Determining the solution space of ANNs is a vital step toward AI in safely critical applications. Since the solution space of an ANN correlates with the size of an ANN, it must be researched how the resulting n-dimensional solution space can be reliably computed and represented. Functional analysis, especially the Hilbert space, might be capable of describing all possible states of the used ANN architecture. Thus, functional safety assessments of AI-based CLCS can be performed systematically and not by a very high number of numerical test cases covering possible corner cases.

6.3. Further Research Directions

Future research directions can be identified by considering the pipeline of applying AI in Figure 2 and the process of AI-empowered CLCS design in Figure 7. Next to the two abovementioned primary research directions, this subsection will summarize further three minor directions. Note that, as stated in Section 2, many research and development efforts are necessary to bridge the gap between control system engineering and AI.

6.3.1. Datasets for AI-Based CLCS

For AI-based CLCSs, ensuring that the recorded process data cover variances without biases is crucial. Therefore, research should focus on systematically recording such data by employing specific stimuli on mechatronic actuators. In addition, creating synthetic, partially synthetic, and hybrid datasets for CLCSs has to be investigated.

6.3.2. Trust and Confidence in AI

Building and measuring trust and confidence in AI [49,50] is quite a new field in AI. Developing interactive HMI to highlight each layer's impact on AI's decision making will help gain the end users' trust and confidence.

6.3.3. Software Solutions for AI-Based CLCS

Easy-to-use software for AI-based CLCS development and testing will help to spread AI-based CLCSs in the field of control system engineering. Software with an interactive HMI, covering all steps shown in Figure 7, from recording the dataset via training the ANN on the control objective to validating the CLCS in the application case, has to be developed. Programming languages like Julia are a good starting point, but control systems engineers are not programmers, so intuitive HMI and UI are needed for AI-based CLCS development.

6.4. Conclusions

Soon, AI-assisted tuning of static parameter sets will also be a promising way to take advantage of AI. However, due to safety considerations, online changes to controller parameters based on AI are not yet suitable for most processes. Similarly, entirely AI-based controllers currently face limitations in terms of functional safety. For these cases, more research is necessary. Nonetheless, granting these AI-based controllers a defined impact on control values, as depicted in Figure 9b, provides a valuable starting point for studying the advantages and disadvantages of AI-based controllers within deployed CLCSs.

Bridging the gap between AI and CLCSs will create an important impetus for innovative control systems designs. AI in the CLCS domain will combine decades of knowledge with process information in the form of recorded data. AI will not replace the experiential knowledge required to develop control systems, but AI will heighten the importance of systematically recording process data, leading into datasets for improving control system design. With upcoming AI-empowered and -based CLCSs, all stakeholders like developers, testers, and end users will gain more and more trust in these systems.

Author Contributions: Conceptualization, J.S. and H.-J.P.; Methodology, J.S.; Validation, H.-J.P.; Formal analysis, J.S. and H.-J.P.; writing—original draft preparation, J.S.; writing—review and editing, J.S. and H.-J.P.; visualization, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the data presented are available within this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Skogestad, S.; Postlethwaite, I. *Multivariable Feedback Control*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2005.
2. Chin, C.S. *Computer-Aided Control Systems Design*; CRC Press: Boca Raton, FL, USA, 2017. [CrossRef]
3. Schöning, J.; Riechmann, A.; Pfisterer, H.J. AI for Closed-Loop Control Systems. In Proceedings of the 2022 14th International Conference on Machine Learning and Computing (ICMLC), Guangzhou, China, 18–21 February 2022. [CrossRef]
4. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the International Conference on Machine Learning (IMCL), Lille, France, 6–11 July 2015; Volume 37, pp. 448–456. [CrossRef]
5. Chen, X.; Liang, C.; Huang, D.; Real, E.; Wang, K.; Liu, Y.; Pham, H.; Dong, X.; Luong, T.; Hsieh, C.J.; et al. Symbolic Discovery of Optimization Algorithms. *arXiv* **2023**, arXiv:2302.06675. [CrossRef]
6. Richter, M.L.; Schöning, J.; Wiedenroth, A.; Krumnack, U. Should You Go Deeper? Optimizing Convolutional Neural Network Architectures without Training. In Proceedings of the International Conference On Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 13–16 December 2021. [CrossRef]
7. Richter, M.L.; Schöning, J.; Wiedenroth, A.; Krumnack, U. Receptive Field Analysis for Optimizing Convolutional Neural Network Architectures without Training. In *Deep Learning Applications*; Springer Nature: Singapore, 2022; Volume 4, pp. 235–261. [CrossRef]
8. Schöning, J.; Westerkamp, C. AI-in-the-Loop—The impact of HMI in AI-based Application. *arXiv* **2023**, arXiv:2303.11508. [CrossRef]
9. Atherton, D.P.; Majhi, S. Limitations of PID controllers. In Proceedings of the American Control Conference, San Diego, CA, USA, 2–4 June 1999. [CrossRef]
10. Åström, K.J.; Hägglund, T. The future of PID control. *Control Eng. Pract.* **2001**, *9*, 1163–1175. [CrossRef]
11. Sung, S.W.; Lee, I.B. Limitations and Countermeasures of PID Controllers. *Ind. Eng. Chem. Res.* **1996**, *35*, 2596–2610. [CrossRef]
12. Rackauckas, C.; Ma, Y.; Martensen, J.; Warner, C.; Zubov, K.; Supekar, R.; Skinner, D.; Ramadhan, A.; Edelman, A. Universal Differential Equations for Scientific Machine Learning. *arXiv* **2021**, arXiv:2001.04385.
13. Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V.B. Julia: A fresh approach to numerical computing. *SIAM Rev.* **2017**, *59*, 65–98. [CrossRef]
14. Rackauckas, C.; Edelman, A.; Fischer, K.; Innes, M.; Saba, E.; Shah, V.B.; Tebbutt, W. Generalized Physics-Informed Learning through Language-Wide Differentiable Programming. In Proceedings of the AAAI Spring Symposium: MLPS 2020, Stanford, CA, USA, 23–25 March 2020.
15. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
16. Raissi, M.; Yazdani, A.; Karniadakis, G.E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* **2020**, *367*, 1026–1030. [CrossRef]
17. Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 3932–3937. [CrossRef] [PubMed]
18. The MathWorks, Inc. Simulik. 2023. Available online: https://www.mathworks.com/products/simulink.html (accessed on 14 August 2023).
19. ESI Group. Xcos. 2023. Available online: https://www.scilab.org/software/xcos (accessed on 14 August 2023).

20. Octave Forge Community. Octave Forge—The 'Control' Package. 2023. Available online: https://octave.sourceforge.io/control (accessed on 14 August 2023).
21. *ISO 26262:2011*; (All Parts): Road Vehicles—Functional Safety. ISO: Geneva, Switzerland, 2011.
22. Kohn, S.C.; de Visser, E.J.; Wiese, E.; Lee, Y.C.; Shaw, T.H. Measurement of Trust in Automation: A Narrative Review and Reference Guide. *Front. Psychol.* **2021**, *12*, 604977. [CrossRef] [PubMed]
23. Gold, C.; Körber, M.; Hohenberger, C.; Lechner, D.; Bengler, K. Trust in Automation–Before and After the Experience of Take-over Scenarios in a Highly Automated Vehicle. *Procedia Manuf.* **2015**, *3*, 3025–3032. [CrossRef]
24. Yang, X.J.; Unhelkar, V.V.; Li, K.; Shah, J.A. Evaluating Effects of User Experience and System Transparency on Trust in Automation. In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, Vienna, Austria, 6–9 March 2017. [CrossRef]
25. Serva, M.A.; Fuller, M.A.; Mayer, R.C. The reciprocal nature of trust: A longitudinal study of interacting teams. *J. Organ. Behav.* **2005**, *26*, 625–648. [CrossRef]
26. Merritt, S.M.; Ilgen, D.R. Not All Trust Is Created Equal: Dispositional and History-Based Trust in Human-Automation Interactions. *Hum. Factors J. Hum. Factors Ergon. Soc.* **2008**, *50*, 194–210. [CrossRef]
27. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
28. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Neural Operator: Graph Kernel Network for Partial Differential Equations. *arXiv* **2020**, arXiv:2003.03485v1.
29. Li, Z.; Zheng, H.; Kovachki, N.; Jin, D.; Chen, H.; Liu, B.; Azizzadenesheli, K.; Anandkumar, A. Physics-Informed Neural Operator for Learning Partial Differential Equations. *arXiv* **2021**, arXiv:2111.03794v1.
30. Somefun, O.A.; Akingbade, K.; Dahunsi, F. The dilemma of PID tuning. *Annu. Rev. Control* **2021**, *52*, 65–74. [CrossRef]
31. Memon, F.; Shao, C. An Optimal Approach to Online Tuning Method for PID Type Iterative Learning Control. *Int. J. Control. Autom. Syst.* **2020**, *18*, 1926–1935. [CrossRef]
32. Liu, J.; Li, T.; Zhang, Z.; Chen, J. NARX Prediction-Based Parameters Online Tuning Method of Intelligent PID System. *IEEE Access* **2020**, *8*, 130922–130936. [CrossRef]
33. Voulgaris, P.G. Control of nested systems. In Proceedings of the American Control Conference ACC, Chicago, IL, USA, 28–30 June 2000. [CrossRef]
34. Lee, Y.; Park, S.; Lee, M. PID Controller Tuning To Obtain Desired Closed Loop Responses for Cascade Control Systems. *Ind. Eng. Chem. Res.* **1998**, *37*, 1859–1865. [CrossRef]
35. Jeng, J.C.; Lee, M.W. Simultaneous automatic tuning of cascade control systems from closed-loop step response data. *J. Process Control* **2012**, *22*, 1020–1033. [CrossRef]
36. Gu, Q.; Dai, Q. A novel active multi-source transfer learning algorithm for time series forecasting. *Appl. Intell.* **2020**, *51*, 1326–1350. [CrossRef]
37. Ye, R.; Dai, Q. Implementing transfer learning across different datasets for time series forecasting. *Pattern Recognit.* **2021**, *109*, 107617. [CrossRef]
38. Duarte, J.; Han, S.; Harris, P.; Jindariani, S.; Kreinar, E.; Kreis, B.; Ngadiuba, J.; Pierini, M.; Rivera, R.; Tran, N.; et al. Fast inference of deep neural networks in FPGAs for particle physics. *J. Instrum.* **2018**, *13*, P07027. [CrossRef]
39. Keras Team. Keras Applications. 2023. Available online: https://keras.io/api/applications/ (accessed on 14 August 2023).
40. Schöning, J. Interactive 3D Reconstruction: New Opportunities for Getting CAD-ready Models. In *2015 Imperial College Computing Student Workshop (ICCSW 2015)*; Schloss Dagstuhl-Leibniz-Zentrum für Informatik: Wadern, Germany, 2015. [CrossRef]
41. Schoning, J.; Richter, M.L. AI-Based Crop Rotation for Sustainable Agriculture Worldwide. In Proceedings of the 2021 IEEE Global Humanitarian Technology Conference (GHTC), Seattle, WA, USA, 19–23 October 2021. [CrossRef]
42. van der Velden, B.H.; Kuijf, H.J.; Gilhuijs, K.G.; Viergever, M.A. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Med. Image Anal.* **2022**, *79*, 102470. [CrossRef]
43. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *Computer Vision—ECCV 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833. [CrossRef]
44. Fong, R.C.; Vedaldi, A. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
45. Zöldy, M.; Szalay, Z.; Tihanyi, V. Challenges in Homologation Process of Vehicles with Artificial Intelligence. *Transport* **2020**, *35*, 435–446. [CrossRef]
46. Athavale, J.; Baldovin, A.; Paulitsch, M. Trends and Functional Safety Certification Strategies for Advanced Railway Automation Systems. In Proceedings of the International Reliability Physics Symposium (IRPS), Dallas, TX, USA, 28 April–30 May 2020. [CrossRef]
47. Tschurtz, H.; Gerstinger, A. The Safety Dilemmas of Autonomous Driving. In Proceedings of the Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 26–27 May 2021. [CrossRef]
48. Ras, G.; Xie, N.; van Gerven, M.; Doran, D. Explainable Deep Learning: A Field Guide for the Uninitiated. *arXiv* **2020**, arXiv:2004.14545v2.

49. Chong, L.; Zhang, G.; Goucher-Lambert, K.; Kotovsky, K.; Cagan, J. Human confidence in artificial intelligence and in themselves: The evolution and impact of confidence on adoption of AI advice. *Comput. Hum. Behav.* **2022**, *127*, 107018. [CrossRef]
50. Babic, B.; Gerke, S.; Evgeniou, T.; Cohen, I.G. Beware explanations from AI in health care. *Science* **2021**, *373*, 284–286. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.