mathematics

MDPI

Review

Review on System Identification, Control, and Optimization Based on Artificial Intelligence

Pan Yu [1,2,*], Hui Wan [2], Bozhi Zhang [2], Qiang Wu [2], Bohao Zhao [2], Chen Xu [2] and Shangbin Yang [2]

[1] School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China
[2] Beijing Institute of Artificial Intelligence, Beijing University of Technology, Beijing 100124, China; huiwan@emails.bjut.edu.cn (H.W.); zhangbozhi@emails.bjut.edu.cn (B.Z.); wuqiang@emails.bjut.edu.cn (Q.W.); zhaobohao@emails.bjut.edu.cn (B.Z.); chenxu@emails.bjut.edu.cn (C.X.); yangshangbin@emails.bjut.edu.cn (S.Y.)
* Correspondence: panyu@bjut.edu.cn

Abstract: Control engineering plays an indispensable role in enhancing safety, improving comfort, and reducing fuel consumption and emissions for various industries, for which system identification, control, and optimization are primary topics. Alternatively, artificial intelligence (AI) is a leading, multi-disciplinary technology, which tries to incorporate human learning and reasoning into machines or systems. AI exploits data to improve accuracy, efficiency, and intelligence, which is beneficial, especially in complex and challenging cases. The rapid progress of AI facilitates major changes in control engineering and is helping advance the next generation of system identification, control, and optimization methods. In this study, we review the developments, key technologies, and recent advancements of AI-based system identification, control, and optimization methods, as well as present potential future research directions.

Keywords: artificial intelligence (AI); control engineering; model prediction control; optimization; parameter estimation; reinforcement learning; system identification; neural networks

MSC: 93-02

check for updates

Academic Editor: Florin Leon

Received: 17 January 2025
Revised: 8 March 2025
Accepted: 11 March 2025
Published: 13 March 2025

Citation: Yu, P.; Wan, H.; Zhang, B.; Wu, Q.; Zhao, B.; Xu, C.; Yang, S. Review on System Identification, Control, and Optimization Based on Artificial Intelligence. *Mathematics* **2025**, *13*, 952. https://doi.org/10.3390/math13060952

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Figuring out models from observations and unearthing their properties is really what science is about, and system identification is an important object of this process. System identification deals with the problem of figuring out mathematical models (such as deterministic models, stochastic models, non-parametric models, and parametric models) of dynamical systems based on the observed input-output data [1].

Although the term system identification was introduced by Lotfi Zadeh in the late 1950s [2], some progress (such as continuous functional [3], analytic functional [4]) was made by Maurice Fréchet and Vito Volterra before then. Since then, system identification has become an established research area within automatic control [5]. Considerable research has been conducted on this. Integral models can be used to describe the evolution of nonlinear dynamical systems [6]. Volterra series can be used to identify nonlinear dynamical systems [7], where integral equations play a key role in system analysis [8]. Moreover, it has been revealed that by using polynomial kernels, Volterra series can be represented implicitly as elements of a reproducing kernel Hilbert space [9]. The convex majorants method can be used to construct main solutions to nonlinear Volterra Equations [10].

A method was developed to compute the solution boundary of an interval outside which blow-up phenomena occur [11].

It is well known that when the essence, basic principles, or fundamental laws are known, mechanistic models (also called first-principles models, analytical models, or knowledge-based models) can be derived for dynamical systems. However, when dynamical systems are too complex to be understood at a fundamental level, data-driven approaches are a highly practical alternative to the first-principles approaches, which leverage computational algorithms or mathematical statistics to identify patterns in input-output data.

In general, the mechanism of dynamical systems is clear according to mechanistic models, while data-driven models are opaque because they are largely developed using mathematical methods. For this reason, mechanistic models are termed white-box models and data-driven models as black-box models. Integrating the traits of both modeling approaches yields hybrid modeling [12], which can be viewed as gray-box models. The structure of a hybrid model usually contains one part derived from first principles and the other part derived from data.

In addition to describing system dynamics, there may be other requirements, including the active manipulation of system behaviors. Control engineering involves applying control methods to design systems such that the systems behave in a desired manner, which has a wide range of applications, including aerospace [13], manufacturing [14], automotive [15], and robotics [16], where precision and reliability are imperative.

There are two main types of control methods, namely, open-loop control and closed-loop control. For simplicity, open-loop control imposes certain preplanned control logic to manipulate the systems. There is no feedback from the states or outputs of the systems. Closed-loop control uses sensors to measure the systems and then make decisions based on how the systems are responding, which allows one to deal with uncertainty and error by introducing feedback. Many methodologies have been developed for feedback control system design.

Mathematical models can be viewed as an interface between a real-world phenomenon and a mathematical world of control theory, which can be used to achieve specific control tasks by control methods. For the period of classical and/or modern control theory, model-based control is the dominant paradigm, and the framework of its control system design is shown in Figure 1. In general, mathematical models are only approximations of real plants, which cannot capture all the features. For example, high-frequency unmodeled dynamics like residual vibration modes were not considered in robotic and spacecraft models [17]. Therefore, although a model-based controller can be designed to achieve satisfactory performance in a simulation environment, it does not necessarily satisfy the predefined requirements for the real plant in practical applications due to various modeling errors or invalid prior assumptions.

Data-driven control is a promising alternative approach to control system design, using the input-output data of a plant to directly design the controller. It circumvents the necessity of system identification and has proliferated rapidly in recent decades. Consequently, data-driven control approaches are basically model-free designs, which can greatly alleviate the theory-practice gap.

Optimization refers to the problem of choosing a set of arguments that maximize or minimize a cost function or multiple cost functions. In the control community, arguments are usually related to control inputs, control parameters, etc., and a cost function or cost functions are related to the performance or requirements of a control system, like estimation errors and tracking errors. Due to various uncertainties in system dynamics, analytical solutions usually cannot be obtained for the formulated optimization problem. Various numerical optimization methods can be used to derive numerical solutions.

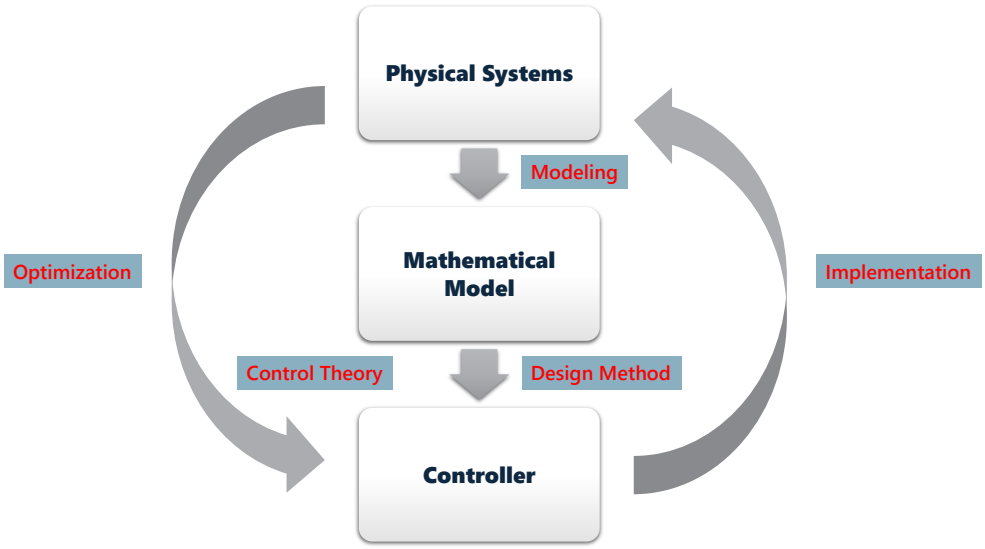


Figure 1. Framework for model-based control system design.

Alternatively, as one of the most highlighted research areas during the past decades, artificial intelligence (AI) is expanding rapidly [18,19], aiming to enable the use of systems or machines with the capability of mimicking human learning and reasoning. Much progress has been made with the help of AI technology in diversified industrial fields, including natural language processing [20], pattern recognition [21], autonomous cars [22], computer vision [23], robotics [24], etc. As the access to the data information of various sensors continues to expand and computing power becomes increasingly accessible, AI-aided or AI-based approaches have gained significant momentum. System identification, control, and optimization benefit greatly from the development of AI.

In this paper, a literature review of AI-based system identification, control, and optimization methods together with their applications and research challenges are presented. The paper is arranged as follows. The review of AI-based system identification methods is presented in Section 2. Section 3 reviews the control methods by integrating AI technology. Optimization methods are reviewed in Section 4. Section 5 presents the challenges and prospects.

2. AI-Based System Identification

In this section, we discuss parameter estimation, data-driven modeling, and hybrid modeling approaches together. Moreover, the advantages and disadvantages are discussed from a control-oriented point of view.

2.1. Parameter Estimation

To ensure mathematical models from mechanistic or data-driven modeling are capable of reflecting their physical counterparts, the parameters need to be precisely estimated. Parameter estimation usually involves the procedures of collecting data, devising estimation algorithms, and optimizing parameters to maximize the likelihood between the prediction and the observation or to minimize the error between the prediction and the observation. Although physics is able to provide knowledge for parameter estimation, the simultaneous estimation of potentially various parameters poses an exceptional challenge for high-dimensional complex systems. AI opens new avenues for the parameter estimation of dynamical systems [25].

By virtue of computational intractability, machine learning algorithms were used for the parameter estimation of complex systems following a transformation to large-scale inverse problems [26]. The stability to small changes in the data is an important property for solutions. Inverse problems that fail to satisfy this stability condition are called ill-posed as per Hadamard [27]. The strategies to solve inverse problems usually involve statistics, optimization, and regularization. Probabilistic tools (such as Bayesian inference and stochastic regularization) were mathematically introduced to address uncertainty, thus transforming ill-posedness into statistically stable solutions [28]. A neural parameter calibration method was developed to estimate parameter probability densities for large-scale multi-agent models [29]. By combining neural ordinary differential equations with physical prior knowledge, a new parameter estimation method was presented [30], which achieved a robust and accurate parameter estimation with a small sample and high noise condition. Deep learning provides an attractive strategy for solving parameter estimation problems by optimization. Regularization is usually used to avoid the ill-posed problem of parameter estimation for a large number of parameters [31]. By introducing additional constraints, a method was developed to regularize ill-posed problems into well-posed formulations [32]. A DeepBayes-based parameter estimator was devised for stochastic nonlinear dynamical models [33].

2.2. Data-Driven Modeling

The increasing complexity of dynamical systems has rendered mechanistic modeling gradually insufficient to meet the requirements due to its inherent limitations. Data-driven modeling does not rely on any prior knowledge or theoretical assumptions. It automatically learns and extracts features from a large amount of data through algorithms to establish models, which makes up for the deficiencies of mechanistic modeling. Therefore, it is increasingly favored in studies and more widely used in various industries.

2.2.1. Machine Learning

The main task of machine learning is to guide computers to learn from data and then use that experience to improve their own performance without explicit programming. In data-driven modeling, as a main learning mode of machine learning, supervised learning algorithms aim to establish the mapping and functional relationships between the inputs and outputs implicitly, with the training dataset consisting of input-and-output pairs.

Neural networks (NNs), support vector machines, logistic regression, etc., and they are supervised learning techniques with shallow structures. NNs were used to predict urban noise and to simulate car tailgating models, respectively [34,35]. Owing to the high approximation capability and the lower reliance on prior knowledge, NNs can handle more variables than traditional mathematical models, and they can deal with the nonlinear factors of dynamical systems. Due to their chaotic, unstable, or time-varying behaviors, it is more challenging to identify a dynamical system for which the observed data is nonstationary. A review about parametric system identification using neural networks was reported by [36], where the model interpretability of three types of NNs (i.e., feedforward NNs, recurrent NNs and encoder–decoder models) was summarized in modeling nonstationary time series. Support vector machines were used to model nonlinear systems [37], which reduced the impact of sample size and noise on system modeling. Furthermore, a bilinear logistic regression approach was proposed for smart grid fault modeling [38], where the parameters were trained using a real-world power grid failure dataset.

2.2.2. Deep Learning

As a breakthrough in AI, deep learning allows for the automatic processing of data by multi-level concatenation to refine highly nonlinear and complex features. Different deep learning based modeling techniques and architectures are used for system identification.

Feed-forward neural networks (FFNNs) are widely used. FFNN training entails random parameter initialization, gradient computation via back-propagation (BP) algorithms, and iterative parameter updates using stochastic gradient descent to minimize the loss function and optimize model performance. This was used to obtain the total nitrogen prediction model of effluents, with significantly improved prediction accuracy [39]. Recurrent neural networks (RNNs) have directed topological connections between neurons for learning features from sequential data [40], which were used for modeling the forward dynamics of soft robots [41]. Long and short-term memory networks are an improved version of RNNs, which were designed to solve the gradient vanishing problem and deal with complex time-series data [42]. Long and short-term memory networks were proposed to predict the acceleration distribution in car-following behavior [43]. Convolutional neural networks (CNNs) especially excel in the field of image recognition [44], where feature learning is achieved by stacking convolutional layers and pooling layers.

2.2.3. Transfer Learning

It is clear that a vast amount of data used for training is necessary for deep learning technology. When it is difficult to find sufficient training data, transfer learning is a good alternative, as it uses models trained from other data sources with certain modifications and improvements for similar fields.

Most data-driven approaches use online data to retrain models. However, unnecessary data updates and retraining are often costly and time consuming. When the studied wind–heat hybrid power system faced the issue of limited input-output data in the current scenario, transfer learning was proposed to utilize the data collected in the historical scenario to construct an online aggregated system frequency response model of the wind–heat hybrid power system operating in the current scenario [45]. A dynamic multi-source transfer learning model was proposed to effectively utilize rather than discard knowledge of historical data under similar working conditions or equipment to solve the defect of the neural network that was difficult to extrapolate [46]. Moreover, transfer learning was used in the initial feature extraction phase, where features were extracted by a pre-trained model, enabling the model to capture complex patterns in videos and extract deep features from video frames [47].

2.3. Hybrid Modeling

The idea of combining mechanistic modeling with data-driven models has emerged in the 1990s [48–50]. Since then, a significant amount of research has been published using the term hybrid modeling in process engineering-related research fields and gray-box modeling in the control and automation fields. Advantages with respect to modeling complex, high-dimensional, and/or computationally expensive systems could greatly benefit from hybrid modeling techniques. Hybrid modeling methodologies are reviewed from the aspect of the structure or typology of hybrid models. The relation between mechanistic modeling and data-driven modeling is shown in Figure 2.

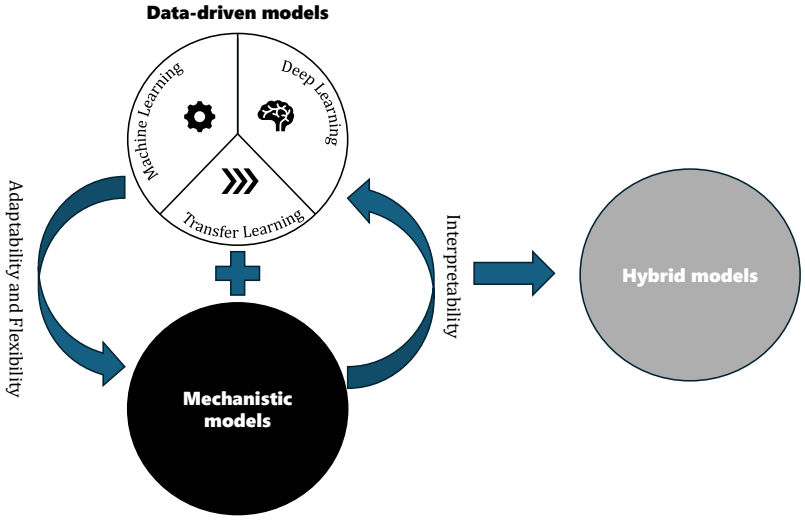


Figure 2. System identification methods.

2.3.1. Parallel Structures

When the performance of a mechanistic model is unsatisfactory due to oversimplifications, unreasonable assumptions, and unmodeled effects [51], hybrid models with parallel structures are usually appropriate. More specifically, mechanistic and data-driven models are executed simultaneously, and then, their outputs are combined in a way (e.g., voting and weighted averaging). It is well known that a mechanistic model represents the dynamics independently of a data-driven model. Consequently, a hybrid model serves to improve the quality of the predictions of the mechanistic model. Moreover, the performance of a hybrid model of a parallel structure depends on the quality of the data-driven model.

Parallel structures are able to reduce the potential bias of a single model by combining the outputs of different models. For example, a hybrid model of a mechanistic model and an FFNN for the wastewater treatment process worked well in parallel, where the output of the mechanistic model served as an input to the neural network and compensated for the model prediction error [52]. A multi-stage model consisting of a mechanistic model and a bi-directional gated recurrent unit model was developed [53], where the outputs of both models were passed by a dense neural network regressor to generate the final predictions.

Another common scenario where a parallel structure is widely used is to embed physical properties into loss functions. The loss function of a traditional data-driven model emphasizes the reduction of the loss between the estimate and the actual observation. However, it ignores the inherent physical properties, which could result in a model that is not consistent with the physics. Physics-informed loss functions are developed to address this challenge. They are constructed to constrain data-driven models to physically consistent spaces [54]. For example, Bayesian optimization algorithms were used to adjust the weighting factors in a loss function, thus optimizing model accuracy [55,56].

2.3.2. Serial Structures

Data-driven models can be advantageous when a mechanistic model is unavailable or too complex to be described mathematically [57]. Therefore, a serial structure can be chosen when a mechanistic model cannot fully map the relation between inputs and outputs because of the complexity, where the mechanistic and data-driven models are executed sequentially with the output of one model serving as the input of the other model. The performance of hybrid models of serial structures depends on how well a mechanistic model represents real phenomena. When the mechanistic model is reliable, the predictive properties of serial structures are usually better than those of parallel structures. However,

parallel structures have better generalization because first-principles knowledge usually generalizes better than pure data.

A physically guided CNN was developed for the fault diagnosis of rolling element bearings [58], where the output of a mechanistic model was used as the input of a data-driven model. Deformation data were obtained by finite element simulation under different geometric gaps, and then, an FFNN prediction model was constructed for the deformation of a typical weld structure [59].

Physical information neural networks (PINNs) are a new type of networks that embed physical principles as a priori knowledge into the training process of NNs. To enable the NNs to fit the data while satisfying physical constraints, some strategies and techniques for incorporating the underlying first principles of the power grid into different types of machine learning approaches were reviewed [60]. PINNs are particularly well suited for scientific computing and engineering problems that involve solving partial differential equations or ordinary differential equations. Some newly developed techniques to enhance PINN performance while reducing training costs have been reviewed [61]. A PINN-based approach was proposed for modeling friction-inclusive dynamics of industrial robots [62], which ensured modeling accuracy while avoiding reliance on joint torque component labels and enhancing its ability to learn nonlinear features.

3. AI-Based Control Methods

The combination of AI and feedback control is an emerging research area, aiming to achieve more efficient and intelligent control systems by integrating the learning and decision-making capabilities of AI with the stability and accuracy of feedback control. The prevailing control methods have been studied extensively with the aid of AI.

3.1. Neural Network Control

Owing to universal approximation, parallel distributed processing, hardware implementation, learning and adaptation, and multivariable systems [63], neural networks are becoming particularly attractive and promising in control system design. Many neural networks with different structures have been introduced, like FFNNs [64–68], radial basis function neural networks (RBFNNs) [69–74], RNNs [75–78], and CNNs [79,80]. The key or challenge of using a neural network for feedback control is to select a suitable control system structure and then demonstrate how the weights of neural networks can be tuned to ensure the control performances (including stability, robustness, dynamic performance, steady-state performance, etc.) of the developed closed-loop feedback systems. Some prevailing control approaches are reviewed in the following.

3.1.1. Neural Network-Based Adaptive Control

Conventional adaptive control without neural networks is usually vulnerable to uncertainties or unknown dynamics and is challenging to deal with complex nonlinear systems. Neural network-based adaptive control is able to directly approximate unknown dynamics using neural networks, which removes the need for precise mathematical models. This is also the main reason why neural network-based adaptive control is especially suitable for complex nonlinear systems or uncertain dynamical systems.

Although promising results can be obtained for the neural network-based adaptive control methods, the closed-loop stability analysis is a burdensome task. Therefore, most of the existing neural network-based adaptive control methods rely on the Lyapunov direct method to guarantee stability. More specifically, a suitable Lyapunov candidate function is chosen first, which relates to both the neural network weights and tracking/estimation errors. Then, the update rules for the network weights are derived such that the time

derivative of the Lyapunov candidate function becomes negative definite or negative semi-definite [81]. By the implicit function theorem, an observer-based direct adaptive fuzzy neural control scheme was proposed for non-affine nonlinear systems with unknown nonlinear structures [82], for which fuzzy neural networks were used to estimate the unknown nonlinear dynamics of the error system while ensuring the boundedness of all system signals by the strictly positive real Lyapunov theory. Furthermore, for highly nonlinear systems, with an online update of the layer width (e.g., neuron numbers) of the neural networks, a direct adaptive state feedback controller was used to approximate and adaptively compensate for unknown nonlinearities [83], whereby the uniform asymptotic stability of tracking errors were guaranteed via the incorporation of robust control terms.

3.1.2. Neural Network-Based Sliding-Mode Control

Owing to its robustness against structured and unstructured uncertainties, as well as its ease of implementation, sliding-mode control (SMC) has gained significant attention in the nonlinear systems control community. However, prior knowledge of the upper bounds of uncertainties or disturbances is usually required for conventional SMC methods, which may be impractical for some applications. Moreover, a large switching gain could bring about significant chattering. Therefore, numerous attempts have been made to integrate neural networks to make SMC more intelligent [84,85]. These can be mainly categorized into two types, namely, model-based and model-free ones.

Based on a mathematical model that may be inaccurate, some neural network-based adaptive SMC methods were designed for robots [86–88], vehicles [89,90], and aircraft [74], where the neural network was used to compensate for model uncertainties or other nonlinearities. Compared with the conventional SMC, the developed method was able to reduce chattering and improve control performance. Alternatively, to remove the prior requirement of an upper bound of uncertainties and disturbances, rather than estimating the real value, neural networks were utilized to estimate the bounds [73,91–94].

For model-free control, neural networks are designed to estimate the whole dynamics of control systems. Although there is no need to calculate mathematical models for model-free control methods, the computation complexity is challenging. For example, a recurrent fuzzy wavelet neural network was integrated into robust adaptive SMC for industrial robot manipulators [95]. An RBFNN was combined with fuzzy sliding mode to tackle the trajectory tracking control [70].

3.1.3. Neural Network-Based Backstepping Control

It is well known that it is challenging to find a suitable control Lyapunov function for a general nonlinear control system. However, for nonlinear systems of a strict feedback form, the backstepping control method is a good solution. More specifically, a recursive design procedure is used, which links finding a control Lyapunov function with the design of a feedback controller. Moreover, the global asymptotic stability of strict feedback systems is guaranteed. By integrating neural networks to estimate unknown dynamics, nonlinearities, or virtual control laws, effective control strategies have been developed for nonlinear systems with uncertainties [96,97].

Owing to their universal approximation property, neural networks were employed to construct the nonlinear mapping online in backstepping control schemes. In earlier research, an RBFNN was used for the online approximation of a linear parametric model for a nonlinear system [71]. For the case where obtaining system states was difficult or costly in practice, neural network-based observers were developed to estimate the unavailable states, which broadened the application of backstepping control [98,99]. To remove the problem of the explosion of complexity of the traditional backstepping design, neural

network techniques were combined with the command-filtered control method to achieve a robust and excellent performance [100–102]. Virtual control laws were approximated step-by-step by neural networks and then incorporated in backstepping control design based on the Lyapunov stability method [103,104].

3.1.4. Neural Network-Based Iterative Learning Control

Iterative learning control (ILC) is widely used in practical applications, such as manufacturing [105], robotics [106], and chemical processing [107], where mass production entails repeated operations on an assembly line. It learns error knowledge from the previous iteration to improve the next iteration [108]. However, it is difficult for conventional ILCs to cope with strong nonlinearities and uncertainties in practice. Therefore, neural networks are also incorporated into ILCs to solve practical difficulties and enhance control performance.

An RBFNN was used to automatically adjust the ILC gain of a nonlinear system by a dynamic linearization technique [109]. A generalized regression neural network was developed as an estimator to calculate key parameters, and then an RBFNN was designed to find the control input by iterative training [110]. Two neural networks were used to model a nonlinear process and as a learning controller, respectively, [111]. Moreover, the convergence analysis was conducted for both P-type and D-type learning controllers. A data-driven prediction model was established by a new RBFNN to estimate non-repeating external disturbances along the iterative learning axis [112]. Based on the previous iterative control input and output error, a BP neural network was used to learn system uncertainties and disturbances and then to compensate for them [113]. To solve the zero tracking problem with inherent noise tolerance, an ILC was proposed based on a modified zeroing neural network [114], which can not only theoretically converge to the exact solution of the zero tracking problem but also eliminate the strict assumptions on the initialization conditions of each iteration [115]. Neural networks were used to estimate the nonlinear portion of an ILC output and then predicted the ILC output for multiple tasks [116]. An inverse model-based ILC method was developed for a multiple input and multiple output (MIMO) nonlinear system, for which a neural network was constructed to observe the system [117].

3.1.5. Summary

The adaptation property arises from the many characteristics of neural networks, which include learnable weights and flexible architectures. According to the mechanism of the weight update policies, neural network control methods can be broadly categorized into two types, which can be seen from Figure 3. The weight update policy of the first type is based on BP algorithms [118,119], and the second type utilizes the Lyapunov stability theory [72]. It is important to mention that the BP algorithm is a class of optimization methods based on gradient information, which is one of the core algorithms in neural network training and is very popular in the control community. Besides the gradient-based methods explained in Section 4.1, second-order optimization methods (such as Newton's method [120], quasi-Newton methods [121]), metaheuristic optimization presented in Section 4.2, and Bayesian optimization [122] can be used to train neural networks. Furthermore, the selection of the number of hidden layers and neurons of a neural network is also important. In general, a neural network with more hidden layers and neurons is capable of providing a better control performance at the cost of higher complexity and computational cost.

Alternatively, it is clear that all the controllers mentioned above are based on standard neural networks. There are a few works based on neural ordinary differential equations. For example, a recurrent wavelet first-order neural network was developed for the multistability control of an erbium-doped fiber laser [123]. Similarly, the selection of the order

number of a neural ordinary differential equation is a trade-off between performance and complexity.

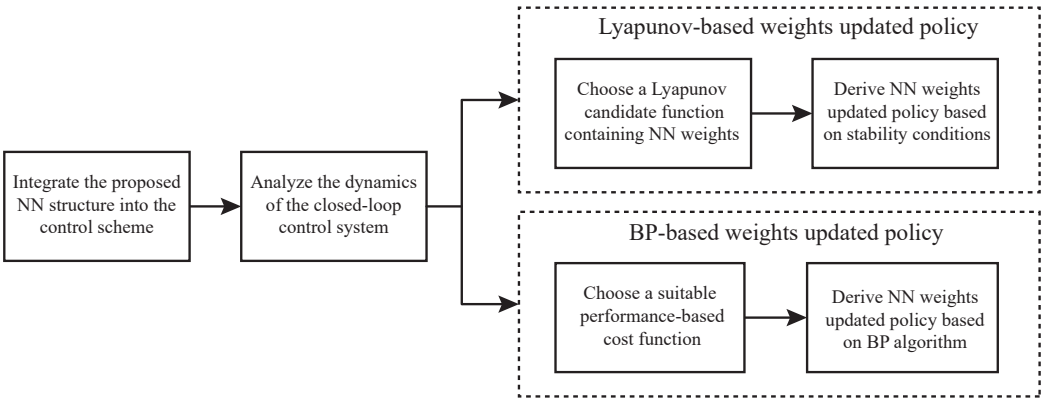


Figure 3. Two types of weight update policies for neural network control.

3.2. Model Prediction Control

Model predictive control (MPC) is also known as receding horizon control, which belongs to the intersection fields of optimization and control. There are several notable advantages for MPC. First, it can seamlessly deal with MIMO systems, which are the main system types in complex industries. Second, various constraints including physical limitations can be well handled for MPC. Third, noting that MPC not only considers the current states and inputs but also takes into account the future states and inputs to achieve optimization control at every sampling time, it is adaptive or robust to future effects. The implementation of MPC includes three key components, namely, model prediction, optimization in a receding horizon, and feedback correction.

However, for dynamical systems requiring high-frequency control, MPC may be infeasible, especially when complex MPC schemes such as nonlinear MPC or robust MPC are used, or when the controller needs to be implemented in a low-cost embedded platform. So, the main challenge of the design and real-time implementation of MPC in practical applications is to have an accurate enough predictive model with a real-time computation capability for large-scale optimization problems. In the past two decades, a great deal of research has been carried out to utilize AI technology to facilitate the design of MPC, for which most works focus on prediction models and fast optimization for high-lever controllers.

3.2.1. Model Prediction

Machine Learning can be used to provide an accurate model at a decreased computational cost and with improved MPC performance [124]. An RNN enjoys the advantages of both high scalability and the capability of modeling multiple-output data. Therefore, RNNs have been widely used in model prediction. For example, to address the limitation of adaptive algorithms based on continuous model update processes and MPC policy redesign, which are constrained to systems with long sampling times, a novel nonlinear system identification and adaptive control method leveraging integrated neural networks was proposed [125]. Addressing at the challenges of nonlinearity, noise, and time delay in industrial processes, and a method of using RNN architectures to solve the problem of modeling and control of complex dynamical systems in industrial processes was proposed [126]. Addressing the challenge of significant hysteresis and unknown state interactions in soft robot systems, a deep-neural network-based method was proposed for modeling and controlling soft robots [127].

A nonlinear auto-regressive predictive model was built for MPC by utilizing a neural network to learn exogenous inputs, and the input-to-state stability of the closed-loop

system was ensured [128]. A path-tracking MPC algorithm was developed for a vision-based mobile robot, where the disturbance was modeled as a Gaussian process [129]. To ensure the safety of the robot when operating under new conditions, a learning probabilistic model was developed by comparing a Gaussian process with a specific form of local linear regression [130]. A novel parameter adaptive MPC architecture was proposed, which enabled online tuning without recomputing large datasets and retraining [131]. To solve the problem related to the fact that the performance of nonlinear MPC models depends on the quality of the underlying model, and that the closed-loop behavior of systems may be difficult to analyze formally, a data-driven economic nonlinear MPC strategy was presented to support parameterization and approximate the action value functions [132].

3.2.2. Fast Optimization

To enable MPC strategies with fast optimization, two types of approaches are developed. The first type includes the development of fast solvers and tailored implementations to calculate formulated optimization problems in real time [133,134]. Machine learning was used to accelerate the original active set to solve the developed quadratic programming-based MPC problem [135], where a state-dependent warm-start was designed by solving the parameterized program offline.

The second type is usually called explicit MPC [136]. With feasibility and stability guarantees, a sampling-based explicit nonlinear MPC was developed for nonlinear systems, for which a support vector machine classifier was used to approximate the feasible region [137]. Neural networks were used to compute the approximate explicit MPC laws, and the feasible control inputs were generated by projecting onto a multifaceted region that was derived from the maximum control invariant set [138]. A deep-learning-based method was developed to efficiently represent an approximation of the MPC laws [133]. Neural networks with modified rectified linear units were constructed to approximate the piecewise affine explicit control laws of MPC, which allowed the approximations to effectively address large-scale problems involving thousands of parameter variables [136].

3.3. Reinforcement Learning Control

As another learning mode of machine learning, unlike supervised learning, which relies on a training dataset with predefined answers and unsupervised learning without desired answers, reinforcement learning (RL) learns a task through a lot of trials. RL is a field closely related to control theory. From the perspective of control theory, the relation of RL with dynamic programming is reviewed first. Through model-based and model-free learning, recent developments in RL in control science are summarized.

3.3.1. Approximate Dynamic Programming

Optimal control is an important branch of control theory, aiming to design controllers to optimize dynamical systems under certain performance metrics [139]. There are three pillars in optimal control, i.e., calculus of variations, Pontryagin's maximum principle, and dynamic programming [140]. Dynamic programming was proposed by Richard Bellman to solve the Bellman equation for a discrete-time system or the Hamilton-Jacobi-Bellman (HJB) for a continuous-time system [141]. However, in practice, directly solving these equations often leads to the curse of dimensionality. In other words, it is computationally large and difficult to realize for high-dimensional systems [140].

To solve this problem, heuristic dynamic programming was developed in 1990, which approximated an optimal value function of the Bellman equation by using approximation functions or models [142]. Then HDP was developed into approximate dynamic programming [143], which is essentially equivalent to the temporal difference methods in RL [144]. In 1989, the proposal of the Q-learning approach combining dynamic program-

ming with online learning marked the emergence of RL in the field of optimal control [145]. As a result, RL and optimal control theory began to integrate, providing a new way to solve complex optimal control problems.

In the context of the optimal control of continuous stochastic processes, the HJB equation and dynamic programming can solve the optimal solution. However, theoretical solutions are difficult to find. Thus, numerical methods such as gradient descent were employed to approximate the optimal solution [146].

3.3.2. Model-Based RL Control

In the field of control systems, with a known system model, the model information can be fully used in the RL algorithm to accelerate the learning process. Given a system model, the value and corresponding reward of each state are accurately calculated, and then, the state transfer probability matrix can be constructed [147]. Using value iteration or policy iteration, the optimal policy can be solved. When the system model is completely known, model-based RL algorithms have been widely used to solve the optimal stabilization problem [148,149], the control problem [150,151], and the tracking problem [152].

3.3.3. Model-Free RL Control

However, in practical applications, it is difficult to obtain an accurate mathematical model for an environment or a dynamic system. Even if an accurate model is obtained, it would be too complex to be used directly. Model-free RL algorithms become particularly important. They perform policy learning through interaction with the environment or system and collect data samples from interaction processes, without relying on an accurate model. There are mainly three categories of model-free RL algorithms [153], including value-based, policy-based, and actor-critic algorithms.

For value-based algorithms, a value function is estimated to evaluate the system states or control actions. An agent or controller selects the optimal policy based on the value function [139]. For example, the Q-learning algorithm is a typical value-based model-free RL algorithm [153]. It is widely used in optimal regulation control [154,155] and optimal tracking control [156,157] for nonlinear systems with unknown models. The deep Q-learning algorithm was used to optimize the energy storage management system of a wind power generator [158].

Unlike value-based algorithms, policy-based algorithms directly learn the mapping relation from states to actions, i.e., the policy function. Instead of explicitly estimating the value function, such methods maximize the long-term cumulative reward by optimizing the policy function [153]. For example, the policy gradient algorithm is a commonly used model-free RL algorithm, which uses gradient descent to optimize parameterized policies [159]. In the gait control for bipedal robots, the policy gradient algorithm learned an optimal gait control strategy in different terrains and motion states by interacting with the environment [160,161].

The actor-critic algorithm combines the advantages of both value and policy functions and learns both value and policy functions to achieve policy optimization. An actor-critic online optimal control algorithm with three neural networks was proposed for the robust control of unknown nonlinear systems [162]. To achieve the optimal synchronous tracking control of heterogeneous multi-agent systems with input-constrained and unknown nonlinearities, an off-policy RL algorithm was proposed, which utilized neural networks to build critic and actor networks, trained them by an actor-critic algorithm, and then obtained the optimal control policy [163]. An actor-critic algorithm was developed to evaluate different control strategies by learning the value function and optimizing the

policy function to achieve an optimal active suspension adjustment to meet the comfort and handling requirements of the vehicle [164].

4. Performance Optimization

Satisfactory performance is one of the end goals of system design or control. Moreover, most system design or control problems can be formulated as optimization tasks. Although gradient-based algorithms are typically recognized because of their computational speed and efficiency, gradients and Hessian matrices are required for the iteration. However, the information is unavailable for black-box cases. Alternatively, when the computational cost is very cheap, the design space is noisy or discontinuous, there may be multiple optima, or when derivatives can not be calculated efficiently, gradient-free optimizers are recommended.

4.1. Gradient-Based Optimization

Gradient-based optimization is the backbone of modern machine learning. It is based on gradient descent and is crucial for training neural networks, deep learning models, and many other machine learning models.

The key idea of gradient descent is that a loss function $J(\theta)$ of parameters θ can be minimized along with the direction of the negative gradient $-\nabla J(\theta)$. In other words, the loss function value can be decreased when θ takes a step in the negative gradient direction. Gradient descent optimizers are based on first-order derivatives, which are more practical than high-order derivative-based algorithms, e.g., Newton's method, for deep models.

Many variants are proposed. The vanilla batch gradient descent algorithm renews the parameters via the gradient of the entire training set, and while simple, it is slow for large datasets and models. Stochastic gradient descent (SGD) addresses this by updating based on the gradient of individual training examples, introducing stochasticity that can help escape suboptimal local minima. In practice, mini-batch SGD that computes the gradient over small subsets of the data is used as a happy medium.

In practice, the gradients are computed by the BP algorithm by utilizing the chain rule. The BP algorithm together with automatic differentiation tools has made gradient-based optimization the default choice for training machine learning models.

4.2. Metaheuristic Optimization

Metaheuristic optimization algorithms are a kind of computational technique widely used in complex optimization problems [165], seeking an approximate optimal solution by imitating the evolution of social behavior or natural phenomena. Moreover, they usually do not rely on the gradient information of the problems considered. Through global or local search, they reach a balance between exploration and development in search space, so as to avoid falling into a local optimal solution. Evolutionary algorithms, particle swarm optimization, ant colony optimization, and simulated annealing are examples of metaheuristic optimization algorithms. Their features are summarized in Table 1.

Genetic algorithms (GAs) are inspired by biological evolution, which simulates natural selection and genetic mechanisms to find optimal solutions. A GA was applied to automatically search for the optimal key hyperparameters of deep RL models [166], such as learning rates, discount factors, and exploration–exploitation trade-offs, which were typically difficult to fine-tune manually. A GA was used for the parameter identification of vapor-liquid thermodynamic models [167]. Differential evolution (DE) is a population-based evolutionary algorithm that guides the optimization process by exploiting the differences between solutions in a population. DE is widely used in the design optimization of power electronic systems, such as the parameter estimation of inverters [168], power converters [169], and solar photovoltaic models [170].

Particle swarm optimization (PSO) methods iteratively solve problems by moving a swarm of particles around the design space until convergence is reached [171,172]. PSO emerges as a powerful optimization technique in fields such as power system state estimation [173]. Ant colony optimization (ACO) is a kind of optimization algorithm inspired by ant foraging behavior, which is mainly used to solve combinatorial optimization problems, such as power scheduling [174], optimal path selection [175], and so on. ACO was used to identify the natural frequencies and damping ratios of mechanical systems [176]. Simulated annealing (SA) solves optimization problems iteratively by simulating the physical annealing process, which is suitable for large-scale, complex nonlinear problems and has been applied in power electronics system design [177] and other fields.

Table 1. Metaheuristic optimization algorithms.

Algorithms	Inspiration	Feature	References
GA	Evolutionary	Robust Global Search	[166,167]
DE	Evolutionary	Efficient Mutation	[168–170]
PSO	Swarm Intelligence	Fast Convergence	[173]
ACO	Swarm Intelligence	Positive Feedback	[177]
SA	Physics	Escape Local Optima	[174,175]

5. Challenges and Prospects

Some future challenges and directions are discussed as follows based on this literature review.

5.1. Challenges

(1) AI models or algorithms usually rely on a large amount of high-quality training data. The means for obtaining these data, especially in dynamic and complex environments, are a significant challenge.
(2) Data-driven techniques for system identification rely on extracted patterns from historical or experimental data. The reasoning mechanism has yet to be fully understood or explained by physics and may thus be opaque to engineers. In other words, interpretability could be a concern due to its black-box nature.
(3) Since the mathematical foundations of AI are yet to be fully established, a practice guide to facilitate the architecture designs and implementation of AI models or algorithms for system identification, control, and optimization is still an open issue.
(4) Although AI models or algorithms can improve the modeling accuracy or control performance of dynamical systems, computational complexity becomes correspondingly more complex. Ways to deal with the trade-off are also a crucial challenge.

5.2. Future Directions

AI is capable of making feedback control loops smarter and more adaptive to uncertainties. Several promising directions to make full use of AI in control systems are given as follows.

(1) More models and learning methods can be used to improve the generalization ability of deep learning. Future efforts should prioritize multi-modal learning architectures that integrate heterogeneous data sources (e.g., multi-modal sensor fusion of vision, LiDAR, and inertial measurement units) to create more comprehensive system representations. Meta-learning approaches, particularly few-shot and zero-shot learning paradigms, could enable control systems to adapt rapidly to novel scenarios with minimal retraining. Foundation-scale models pre-trained on multi-domain physical system datasets demonstrate transformative potential. They can serve as universal

priors for low-level control tasks while retaining domain-specific knowledge through lightweight fine-tuning. For instance, a physics-informed large language model could interpret system dynamics from textual maintenance logs and simultaneously process numerical sensor data, enabling cross-domain knowledge transfer in industrial control engineering applications.

(2) Developing a new, effective, and more interpretable architecture to implement AI models or algorithms in feedback control systems could make the decision-making process of control systems more transparent, ensuring stability and real-time guarantees. Neural ordinary differential equation networks could be synergistically integrated with traditional MPC frameworks, where the neural component learns residual dynamics while the MPC core provides stability guarantees through convex optimization constraints. Furthermore, attention mechanisms and symbolic regression layers should be incorporated into control strategies to produce human-understandable explanations for control actions. Moreover, the integration of performance assessment tools, such as reachability analysis and Lyapunov functional synthesis, with neural network-based controllers will be essential for safety-critical applications like autonomous vehicles and medical robotics.

(3) Exploring strategies regarding the ways machine learning is used to model and handle uncertainties can achieve more robust control. For most mechanical systems, the dominant dynamics can be obtained by first principles. Uncertainties including parameter variations (e.g., the moment of the inertia of robotic arms) and unmodeling dynamics (e.g., the aerodynamic disturbances of unmanned aerial vehicles) are the main causes of performance deterioration. A hierarchical learning framework could be developed where Bayesian neural networks quantify uncertainty distributions, while adversarial reinforcement learning agents train controllers to cope optimally under worst-case uncertainty scenarios. Physics-guided uncertainty propagation methods should be investigated, whereby learned uncertainty bounds are systematically incorporated into robust control strategies such as H infinity control and SMC. For distributed systems, federated learning architectures could enable collaborative uncertainty modeling across fleets of cyber-physical systems while preserving data privacy.

(4) Most works focus on uncertainty estimation and compensation without considering the learning performance. Utilizing AI-based optimization algorithms to calculate the control input could be a more efficient and direct control method. Implicit neural representation networks could parameterize entire families of optimal control laws, enabling real-time solutions for nonlinear MPC problems without iterative computations. Evolutionary strategies enhanced by neural surrogates may discover non-conventional control strategies that outperform conventional PID or linear quadratic Gaussian designs in complex multi-objective scenarios. For large-scale systems, graph neural networks could solve distributed optimization problems by learning message-passing mechanisms between subsystems. Crucially, these approaches must address the duality between learning performance and control stability through novel loss functions that penalize Lyapunov function derivatives or contraction metric violations.

Author Contributions: Conceptualization, P.Y.; methodology, P.Y. and H.W.; software, B.Z. (Bozhi Zhang) and Q.W.; validation, P.Y., H.W. and B.Z.; formal analysis, B.Z. (Bohao Zhao) and B.Z. (Bozhi Zhang); investigation, C.X., S.Y. and B.Z. (Bozhi Zhang); resources, H.W., B.Z. (Bozhi Zhang) and B.Z. (Bohao Zhao); data curation, C.X.; writing—original draft preparation, P.Y. and B.Z. (Bozhi Zhang); writing—review and editing, Q.W. and H.W.; visualization, B.Z. (Bozhi Zhang) and H.W.; supervision, P.Y.; project administration, P.Y.; funding acquisition, P.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the National Natural Science Foundation of China under Grant 62003010 and the Scientific Research Project of the Beijing Educational Committee under Grant KM202410005036.

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

References

1. Ljung, L. Perspectives on system identification. *Annu. Rev. Control* **2010**, *34*, 1–12. [CrossRef]
2. Zadeh, L. On the identification problem. *IRE Trans. Circuit Theory* **1956**, *3*, 277–281. [CrossRef]
3. Fréchet, M. Sur les fonctionnelles continues. *Ann. Sci. L'éCole Norm. SupéRieure* **1910**, *27*, 193–216. [CrossRef]
4. Volterra, V. *Theory of Functionals and of Integral and Integro-Differential Equations*; Dover Publications, Inc.: New York, NY, USA, 1958.
5. Ljung, L. *System Identification—Theory for the User*, 2nd ed.; Prentice-Hall: Hoboken, NJ, USA, 1999.
6. Sidorov, D. *Integral Dynamical Models: Singularities, Signals and Control*; World Scientific: Singapore, 2014.
7. Sidorov, D.; Tynda, A.; Muratov, V.; Yanitsky, E. Volterra Black-Box Models Identification Methods: Direct Collocation vs. Least Squares. *Mathematics* **2024**, *12*, 227.[CrossRef]
8. Apartsyn, A.S. *Nonclassical Linear Volterra Equations of the First Kind*; De Gruyter: Berlin, Germany, 2011.
9. Franz, M.O.; Schölkopf, B. A unifying view of Wiener and Volterra theory and polynomial kernel regression. *Neural Comput.* **2006**, *18*, 3097–3118. [CrossRef]
10. Sidorov, D.; Sidorov, N.A. Convex majorants method in the theory of nonlinear Volterra equations. *Banach J. Math. Anal.* **2012**, *6*, 1–10. [CrossRef]
11. Sidorov, D. Existence and blow-up of Kantorovich principal continuous solutions of nonlinear integral equations. *Differ. Equ.* **2014**, *50*, 1175–1182. [CrossRef]
12. Schweidtmann, A.M.; Zhang, D.D.; Stosch, M.V. Review and perspective on hybrid modeling methodologies. *Digit. Chem. Eng.* **2024**, *10*, 100136. [CrossRef]
13. Chernyshev, S.; Sypalo, K.; Bazhenov, S. Intelligent control algorithms to ensure the flight safety of aerospace vehicles. *Acta Astronaut.* **2025**, *226*, 782–790. [CrossRef]
14. Monostori, L.; Kádár, B.; Bauernhansl, T. Cyber-physical systems in manufacturing. *CIRP Ann.* **2016**, *65*, 621–641. [CrossRef]
15. Liu, F.Y.; Jin, L.; Shang, M.; Wang, X. ACP-Incorporated Perturbation-Resistant Neural Dynamics Controller for Autonomous Vehicles. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4675–4686. [CrossRef]
16. Huang, X.; Rong, Y.; Gu, G. High-Precision Dynamic Control of Soft Robots with the Physics-Learning Hybrid Modeling Approach. *IEEE/ASME Trans. Mechatron.* 2024, *early access*. [CrossRef]
17. Khaki-Sedigh A. *An Introduction to Data-Driven Control Systems*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2023.
18. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
19. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
20. Liu, D. Design Information Extraction and Visual Representation based on Artificial Intelligence Natural Language Processing Techniques. In Proceedings of the International Conference on Computer Vision, Image and Deep Learning, Zhuhai, China, 12–14 May 2023; pp. 154–158.
21. Zimo, Y.; Jian, P.; Ru, W.; Xiang, Y.X. Embrace sustainable AI: Dynamic data subset selection for image classification. *Pattern Recognit.* **2024**, *151*, 110392.
22. Kashyap, A.; Ray, A.; Kalyan, B.; Bhushan, P. A Minimalistic Model for Converting Basic Cars Into Semi-Autonomous Vehicles Using AI and Image Processing. In Proceedings of the 2023 International Conference on Applied Intelligence and Sustainable Computing, Dharwad, India, 16–17 June 2023; pp. 1–6.
23. Bhamare, A.; Upadhyay, V.; Bansal, P. AI based Plant Growth Monitoring System using Computer Vision. In Proceedings of the 2023 IEEE Technology & Engineering Management Conference—Asia Pacific, Bengaluru, India, 14–16 December 2023; pp. 1–5.
24. Jorge, R.; Rui, L.; Tiago, E.; Sara, P. Robotic Process Automation and Artificial Intelligence in Industry 4.0—A Literature review. *Procedia Comput. Sci.* **2021**, *181*, 51–58.
25. Kutz, J. Machine learning for parameter estimation. *Proc. Natl. Acad. Sci. USA* **2023**, *120*, e2300990120. [CrossRef]
26. Richard, C.A.; Brian, B.; Clifford, H.T. *Parameter Estimation and Inverse Problems*, 2nd ed.; Academic Press: Cambridge, MA, USA, 2013; pp. 93–127.
27. Kabanikhin, S.I. *Inverse and Ill-Posed Problems*; de Gruyter: Berlin, Germany, 2008.
28. Lavrentiev, M.M. *Some Improperly Posed Problems of Mathematical Physics*; Springer: Berlin/Heidelberg, Germany, 1967.
29. Gaskin, T.; Pavliotis, G.A.; Girolami, M. Neural parameter calibration for large-scale multi-agent models. *Proc. Natl. Acad. Sci. USA* **2023**, *120*, e2216415120. [CrossRef]
30. Yong, Y.; Haibin, L. Neural ordinary differential equations for robust parameter estimation in dynamic systems with physical priors. *Appl. Soft Comput.* **2025**, *169*, 112649. [CrossRef]

31. Pillonetto, G.; Chen, T.; Chiuso, A.; Nicolao, G.; Ljung, L. *Regularized System Identification: Learning Dynamic Models from Data*; Springer: Cham, Switzerland, 2022.
32. Tikhonov, A.N.; Arsenin, V.Y. *Solutions of Ill-Posed Problems*; Halsted Press: Ultimo, Australia, 1977.
33. Anubhab, G.; Mohamed, A.; Saikat, C.; Håkan, H. DeepBayes—An estimator for parameter estimation in stochastic nonlinear dynamical models. *Automatica* **2024**, *159*, 111327.
34. Genaro, N.; Torija, A.; Ramos, A.; Requena, I.; Ruiz, D.P.; Zamorano, M. Modeling environmental noise using artificial neural networks. In Proceedings of the International Conference on Intelligent Systems Design and Applications, Pisa, Italy, 30 November–2 December 2009; pp. 215–219.
35. Khodayari, A.; Ghaffari, A.; Kazemi, R.; Braunsting, R. A modified car-following model based on a neural network model of the human driver effects. *IEEE Trans. Syst.* **2012**, *42*, 1440–1449. [CrossRef]
36. Dong, A.; Starr, A.; Zhao, Y. Neural network-based parametric system identification: A review. *Int. J. Syst. Sci.* **2023**, *54*, 2676–268. [CrossRef]
37. Zhang, H.R.; Wang, X.S.; Zhang, C.J.; Xu, X.L. Modeling nonlinear dynamical systems using support vector machine. In Proceedings of the International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; pp. 3204–3209.
38. De Santis, E.; Rizzi, A. Modeling failures in smart grids by a bilinear logistic regression approach. *Neural Netw.* **2024**, *174*, 106245. [CrossRef] [PubMed]
39. Zhao, Z.H.; Wang, Z.H.; Yuan, J.L.; Ma, J.; He, Z.L.; Xu, Y.L.; Shen, X.J.; Zhu, L. Development of a novel feedforward neural network model based on controllable parameters for predicting effluent total nitrogen engineering. *Engineering* **2021**, *7*, 195–202. [CrossRef]
40. Hihi, S.E.; Bengio, Y. Hierarchical recurrent neural networks for long-term dependencies. *Int. Conf. Neural Inf. Process. Syst.* **1995**, *8*, 493–499.
41. Thuruthel, T.G.; Falotico, E.; Renda, F.; Laschi, C. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Trans. Robot.* **2019**, *35*, 124–134. [CrossRef]
42. Wang, J.J.; Ma, Y.; Zhang, L.B.; Gao, R.X.; Wu, D.Z. Deep learning for smart manufacturing: Methods and applications. *J. Manuf. Syst.* **2018**, *48*, 144–156. [CrossRef]
43. Morton, J.; Wheeler, T.A.; Kochenderfer, M.J. Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1289–1298. [CrossRef]
44. Yu, M.; Zhang, G.; Li, Q.; Chen, F. PGMM—Pre-trained gaussian mixture model based convolution neural network for electroencephalography imagery analysis. *IEEE Access* **2020**, *8*, 157418–157426. [CrossRef]
45. Zhang, J. Sfr modeling for hybrid power systems based on deep transfer learning. *IEEE Trans. Ind. Inform.* **2024**, *20*, 399–410. [CrossRef]
46. Zhou, X.; Zhai, N.; Li, S.; Shi, H. Time series prediction method of industrial process with limited data based on transfer learning. *IEEE Trans. Ind. Inform.* **2023**, *19*, 6872–6882. [CrossRef]
47. Hassan, N.; Miah, A.; Shin, J. A deep bidirectional LSTM model enhanced by transfer-learning-based feature extraction for dynamic human activity recognition. *Appl. Sci.* **2024**, *14*, 603. [CrossRef]
48. Psichogios, D.C.; Ungar, L.H. A hybrid neural network-first principles approach to process modeling. *AIChE J.* **1992**, *38*, 1499–1511. [CrossRef]
49. Su, H.T.; Bhat, N.; Minderman, P.A.; McAvoy, T.J. Integrating neural networks with first principles models for dynamic modeling. In *Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes*; Elsevier: Amsterdam, The Netherlands, 1993; pp. 327–332.
50. Thompson, M.L.; Kramer, M.A. Modeling chemical processes using prior knowledge and neural networks. *AIChE J.* **1994**, *40*, 1328–1340. [CrossRef]
51. Yang, S.; Pranesh, N.; Sambit, G.; Wayne, B. Hybrid modeling in the era of smart manufacturing. *Comput. Chem. Eng.* **2020**, *140*, 106874. [CrossRef]
52. Chen, Z.; Zhang, J.; Chu, Q.; Wang, Y. Study on hybrid modeling of urban wastewater treatment process. In Proceedings of the 2022 34th Chinese Control and Decision Conference, Hefei, China, 15–17 August 2022; pp. 792–797.
53. Wang, J.J.; Li, Y.L.; Zhao, R.; Gao, R.X. Physics guided neural network for machining tool wear prediction. *J. Manuf. Syst.* **2020**, *57*, 298–310. [CrossRef]
54. Wang, J.J.; Li, Y.L.; Gao, R.X.; Zhang, F.L. Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *J. Manuf. Syst.* **2022**, *63*, 381–391. [CrossRef]
55. Wei, C.J.; Zhang, J.X.; Liechti, M.; Wu, C.L. Data-driven modeling of interfacial traction–separation relations using a thermodynamically consistent neural network. *Comput. Methods Appl. Mech. Eng.* **2023**, *404*, 115826. [CrossRef]
56. Gao, T.; Zhu, H.; Wu, J. Hybrid physics data-driven model-based fusion framework for machining tool wear prediction. *Int. J. Adv. Manuf. Technol.* **2024**, *132*, 1481–1496. [CrossRef]

57. Zendehboudi, S.; Rezaei, N.; Lohi, A. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. *Appl. Energy* **2018**, *228*, 2539–2566. [CrossRef]
58. Sadoughi, M.; Hu, C. Physics-based convolutional neural network for fault diagnosis of rolling element bearings. *IEEE Sens. J.* **2019**, *19*, 4181–4192. [CrossRef]
59. Wang, J.M.; Liu, J.Y.; Hou, X.J. A hybrid mechanics-data modeling approach for predicting the accuracy of train body flexible assembly. *Mech. Syst. Signal Process.* **2024**, *202*, 111242.
60. Jabbari, Z.M.; Chatterjee, P.; Srivastava, A.K. Physics-informed machine learning for data anomaly detection, classification, localization, and mitigation: A review, challenges, and path forward. *J. Mech. Eng.* **2024**, *60*, 177–186.
61. Lawal, Z.K.; Lai, D.T.C.; CheIdris, A. Physics-informed neural network (PINN) evolution and beyond: A systematic literature review and bibliometric analysis. *Big Data Cogn. Comput.* **2022**, *26*, 140. [CrossRef]
62. Hu, H.; Shen, Z.; Zhuang, C. A PINN-based friction-inclusive dynamics modeling method for industrial robots. *IEEE Trans. Ind. Electron.* 2024, *early access*. [CrossRef]
63. Hunt, K.J.; Sbarbaro, D.; Zbikowski, R.; Gawthrop, P.J. Neural networks for control systems—A survey. *Automatica* **1992**, *28*, 1083–1112. [CrossRef]
64. Zhan, X.; Wang, W.; Chung, H. A neural-network-based color control method for multi-color LED systems. *IEEE Trans. Power Electron.* **2019**, *34*, 7900–7913. [CrossRef]
65. Wang, W.; Chung, H.S.; Cheng, R.; Leung, C.S.; Zhan, X.; Lo A.W.; Kwok, J.; Xue, C.J.; Zhang, J. Training neural-network-based controller on distributed machine learning platform for power electronics systems. In Proceedings of the 2017 IEEE Energy Conversion Congress and Exposition, Cincinnati, OH, USA, 1–5 October 2017; pp. 3083–3089.
66. Novak, M.; Dragicevic, T. Supervised imitation learning of finite-set model predictive control systems for power electronics. *IEEE Trans. Ind. Electron.* **2021**, *68*, 1717–1723. [CrossRef]
67. Lin, C.M.; Hung, K.N.; Hsu, C.F. Adaptive neuro-wavelet control for switching power supplies. *IEEE Trans. Power Electron.* **2007**, *22*, 87–95. [CrossRef]
68. Zhao, J.; Bose, B. Neural-network-based waveform processing and delayless filtering in power electronics and AC drives. *IEEE Trans. Ind. Electron.* **2004**, *51*, 981–991. [CrossRef]
69. Lin, F.J.; Teng, L.T.; Yu, M.H. Radial basis function network control with improved particle swarm optimization for induction generator system. *IEEE Trans. Power Electron.* **2008**, *23*, 2157–2169. [CrossRef]
70. Wang, F.; Chao, Z.Q.; Huang, L.B.; Li, H.Y.; Zhang, C.Q. Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode. *Clust. Comput.* **2019**, *22*, 5799–5809. [CrossRef]
71. Polycarpou, M. Stable adaptive neural control scheme for nonlinear systems. *IEEE Trans. Autom. Control* **1996**, *41*, 447–451. [CrossRef]
72. Seshagiri, S.; Khalil, H. Output feedback control of nonlinear systems using RBF neural networks. *IEEE Trans. Neural Netw.* **2000**, *11*, 69–79. [CrossRef] [PubMed]
73. Fei, J.; Ding, H. Adaptive sliding mode control of dynamic system using RBF neural network. *Nonlinear Dyn.* **2012**, *70*, 1563–1573. [CrossRef]
74. Li, S.; Wang, Y.; Tan, J.; Zheng, Y. Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft. *Neurocomputing* **2016**, *216*, 126–134. [CrossRef]
75. Lin, W.M.; Hong, C.M. A new Elman neural network-based control algorithm for adjustable-pitch variable-speed wind-energy conversion systems. *IEEE Trans. Power Electron.* **2011**, *26*, 473–481. [CrossRef]
76. Lin, F.J.; Teng, L.T.; Chu, H. A robust recurrent wavelet neural network controller with improved particle swarm optimization for linear synchronous motor drive. *IEEE Trans. Power Electron.* **2008**, *23*, 3067–3078. [CrossRef]
77. Lin, F.J.; Huang, P.K.; Wang, C.C.; Teng, L.T. An induction generator system using fuzzy modeling and recurrent fuzzy neural network. *IEEE Trans. Power Electron.* **2007**, *22*, 260–271. [CrossRef]
78. Chen, C.S. TSK-type self-organizing recurrent-neural-fuzzy control of linear microstepping motor drives. *IEEE Trans. Power Electron.* **2010**, *25*, 2253–2265. [CrossRef]
79. Bose, B.K. Neural network applications in power electronics and motor drives—An introduction and perspective. *IEEE Trans. Ind. Electron.* **2007**, *54*, 14–33. [CrossRef]
80. Kiranyaz, S.; Gastli, A.; Ben-Brahim, L.; Al-Emadi, N.; Gabbouj, M. Real-time fault detection and identification for MMC using 1-D convolutional neural networks. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8760–8771. [CrossRef]
81. Esfandiari K.; Abdollahi, F.; Talebi, H.A. *Neural Network-Based Adaptive Control of Uncertain Nonlinear Systems*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022.
82. Leu, Y.G.; Wang, W.Y.; Lee, T.T. Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems. *IEEE Trans. Neural Netw.* **2005**, *16*, 853–861. [CrossRef] [PubMed]
83. Park, J.H.; Huh, S.H.; Kim, S.H.; Seo, S.J.; Park, G.T. Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks. *IEEE Trans. Neural Netw.* **2005**, *16*, 414–422. [CrossRef] [PubMed]

84. Wang, S.; Hui, Y.; Sun, X.; Shi, D. Neural network sliding mode control of intelligent vehicle longitudinal dynamics. *IEEE Access* **2019**, *7*, 162333–162342. [CrossRef]
85. Bao, Y.; Thesma, V.; Velni, J.M. Physics-guided and neural network learning-based sliding mode control. *IFAC-PapersOnLine* **2021**, *54*, 705–710. [CrossRef]
86. Sun, T.; Pei, H.; Pan, Y.; Zhou, H.; Zhang, C. Neural network-based sliding mode adaptive control for robot manipulators. *Neurocomputing* **2011**, *74*, 2377–2384. [CrossRef]
87. Yen, V.T.; Nan, W.Y.; Van Cuong, P.; Quynh, N.X.; Thich, V.H. Robust adaptive sliding mode control for industrial robot manipulator using fuzzy wavelet neural networks. *Int. J. Control Autom. Syst.* **2017**, *15*, 2930–2941. [CrossRef]
88. Yen, V.T.; Nan, W.Y.; Van Cuong P. Robust adaptive sliding mode neural networks control for industrial robot manipulators. *Int. J. Control Autom. Syst.* **2019**, *17*, 783–792. [CrossRef]
89. Lu, X.; Zhang, X.; Zhang, G.; Fan, J.; Jia, S. Neural network adaptive sliding mode control for omnidirectional vehicle with uncertainties. *ISA Trans.* **2019**, *86*, 201–214. [CrossRef]
90. Sun, Z.; Zou, J.; He, D.; Zhu, W. Path-tracking control for autonomous vehicles using double-hidden-layer output feedback neural network fast nonsingular terminal sliding mode. *Neural Comput. Appl.* **2021**, *34*, 5135–5150. [CrossRef]
91. Liu, H.; Gong, Z. Upper bound adaptive learning of neural network for the sliding mode control of underwater robot. In Proceedings of the 2008 International Conference on Advanced Computer Theory and Engineering, Phuket, Thailand, 20–22 December 2008; pp. 276–280.
92. Bagheri, A.; Moghaddam, J.J. Simulation and tracking control based on neural-network strategy and sliding-mode control for underwater remotely operated vehicle. *Neurocomputing* **2009**, *72*, 1934–1950. [CrossRef]
93. Tlijani, H.; Jouila, A.; Nouri, K. Wavelet neural network sliding mode control of two rigid joint robot manipulator. *Adv. Mech. Eng.* **2022**, *14*, 16878132221119886. [CrossRef]
94. Boukens, M.; Boukabou, A.; Chadli, M. Robust adaptive neural network-based trajectory tracking control approach for nonholonomic electrically driven mobile robots. *Robot. Auton. Syst.* **2017**, *92*, 30–40. [CrossRef]
95. Yen, V.T.; Nan, W.Y.; Van, C.P. Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators. *Neural Comput. Appl.* **2018**, *31*, 6945–6958. [CrossRef]
96. Lai, G.; Tao, G.; Zhang, Y.; Liu, Z. Adaptive control of noncanonical neural-network nonlinear systems with unknown input dead-zone characteristics. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 3346–3360. [CrossRef] [PubMed]
97. El-Sousy, F.F.M.; Abuhasel, K.A. Adaptive nonlinear disturbance observer using a double-loop self-organizing recurrent wavelet neural network for a two-axis motion control system. *IEEE Trans. Ind. Appl.* **2018**, *54*, 764–786. [CrossRef]
98. Choi, J.Y.; Farrell, J. Adaptive observer backstepping control using neural networks. *IEEE Trans. Neural Netw.* **2001**, *12*, 1103–1112. [CrossRef]
99. Xi, R.D.; Ma, T.N.; Xiao, X.; Yang, Z.X. Design and implementation of an adaptive neural network observer–based backstepping sliding mode controller for robot manipulators. *Trans. Inst. Meas. Control* **2024**, *46*, 1093–1104. [CrossRef]
100. Zhou, Z.; Yu, J.; Yu, H.; Lin, C. Neural network-based discrete-time command filtered adaptive position tracking control for induction motors via backstepping. *Neurocomputing* **2017**, *260*, 203–210. [CrossRef]
101. Zhang, C.; Wang, C.; Wei, Y.; Wang, J. Neural-based command filtered backstepping control for trajectory tracking of underactuated autonomous surface vehicles. *IEEE Access* **2020**, *8*, 42481–42490. [CrossRef]
102. Niu, S.; Wang, J.; Zhao, J.; Shen, W. Neural network-based finite-time command-filtered adaptive backstepping control of electro-hydraulic servo system with a three-stage valve. *ISA Trans.* **2024**, *144*, 419–435. [CrossRef]
103. Zhou, Q.; Zhao, S.; Li, H.; Lu, R.; Wu, C. Adaptive neural network tracking control for robotic manipulators with dead zone. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3611–3620. [CrossRef] [PubMed]
104. Zong, G.; Xu, Q.; Zhao, X.; Su, S.F.; Song, L. Output-feedback adaptive neural network control for uncertain nonsmooth nonlinear systems with input deadzone and saturation. *IEEE Trans. Cybern.* **2023**, *53*, 5957–5969. [CrossRef] [PubMed]
105. Armstrong, A.A.; Alleyne, A.G. A multi-input single-output iterative learning control for improved material placement in extrusion-based additive manufacturing. *Control Eng. Pract.* **1990**, *111*, 104783. [CrossRef]
106. Schwegel, M.; Kugi, A. A simple computationally efficient path ILC for industrial robotic manipulators. In Proceedings of the IEEE International Conference on Robotics and Automation, Yokohama, Japan, 13–17 May 2024; pp. 2133–2139.
107. Hao, S.L.; Liu, T.; Rogers, E. Extended state observer based indirect-type ILC for single-input single-output batch processes with time- and batch-varying uncertainties. *Automatica* **2020**, *112*, 108673. [CrossRef]
108. Bristow, D.A.; Tharayil, M.; Alleyne, A.G. A survey of iterative learning control. *IEEE Control Syst. Mag.* **2006**, *26*, 96–114.
109. Yu, X.; Hou, Z.; Polycarpou, M.M. Controller-dynamic-linearization-based data-driven ILC for nonlinear discrete-time systems with RBFNN. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 4981–4992. [CrossRef]
110. Shi, Q.; Huang, X.; Meng, B. Neural network-based iterative learning control for trajectory tracking of unknown SISO nonlinear systems. *Expert Syst. Appl.* **2023**, *232*, 120863. [CrossRef]
111. Patan, K.; Patan, M. Neural-network-based iterative learning control of nonlinear systems. *ISA Trans.* **2020**, *98*, 445–453. [CrossRef]

112. Yu, Q.; Hou, Z.; Bu, X.; Yu, Q. RBFNN-based data-driven predictive iterative learning control for nonaffine nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 1170–1182. [CrossRef]
113. Chen, Z.; Hou, Y.; Huang, R. Neural network compensator-based robust iterative learning control scheme for mobile robots nonlinear systems with disturbances and uncertain parameters. *Appl. Math. Comput.* **2024**, *469*, 128549. [CrossRef]
114. Zhang, Y.; Ge, S.S. Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans. Neural Netw.* **2005**, *16*, 1477–1490. [CrossRef] [PubMed]
115. Hu, Y. Noise-tolerant ZNN-based data-driven iterative learning control for discrete nonaffine nonlinear MIMO repetitive systems. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 344–361. [CrossRef]
116. Zhang, D.; Wang, Z.; Masayoshi, T. Neural-network-based iterative learning control for multiple tasks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4178–4190. [CrossRef]
117. Lv, Y.; Ren, X.; Tian, J. Inverse-model-based iterative learning control for unknown MIMO nonlinear system with neural network. *Neurocomputing* **2023**, *519*, 187–193. [CrossRef]
118. Abdollahi, F.; Talebi, H.; Patel, R. Stable identification of nonlinear systems using neural networks: Theory and experiments. *IEEE Trans. Mechatron.* **2006**, *11*, 488–495. [CrossRef]
119. Chen, F.C.; Liu, C.C. Adaptively controlling nonlinear continuous-time systems using multilayer neural networks. *IEEE Trans. Autom. Control* **1994**, *39*, 1306–1310. [CrossRef]
120. Wang, C.C.; Tan, K.L.; Chen, C.T.; Lin, Y.H.; Keerthi, S.S.; Mahajan, D.; Sundararajan, S.; Lin, C.J. Distributed Newton Methods for Deep Neural Networks. *Neural Comput.* **2018**, *30*, 1673–1724. [CrossRef]
121. Indrapriyadarsini, S.; Mahboubi, S.; Ninomiya, H.; Kamio, T.; Asai, H. Accelerating Symmetric Rank-1 Quasi–Newton Method with Nesterov's Gradient for Training Neural Networks. *Algorithms* **2022**, *15*, 6. [CrossRef]
122. Guilhoto, L.F.; Perdikaris, P. Composite Bayesian Optimization in function spaces using NEON—Neural Epistemic Operator Networks. *Sci. Rep.* **2024**, *14*, 29199. [CrossRef]
123. Magallón, D.A.; Jaimes, R.R.; García, J.H.; Huerta, C.G.; López, M.D.; Pisarchik, A.N. Control of Multistability in an Erbium-Doped Fiber Laser by an Artificial Neural Network: A Numerical Approach. *Mathematics* **2022**, *10*, 3140. [CrossRef]
124. Norouzi, A.; Heidarifar, H.; Borhan, H.; Shahbakhti, M.; Koch, C.R. Integrating machine learning and model predictive control for automotive applications: A review and future directions. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105878. [CrossRef]
125. Akpan, V.A.; Hassapis, G.D. Nonlinear model identification and adaptive model predictive control using neural networks. *IEEE Robot. Autom. Lett.* **2011**, *4*, 177–194. [CrossRef] [PubMed]
126. Lanzetti, N.; Lian, Y.Z.; Cortinovis, A.; Dominguez, L.; Mercangoz, M.; Jones, C. Recurrent neural network based MPC for process industries. *Eur. Control Conf.* **2019**, *6*, 1005–1010.
127. Gillespie, M.T.; Best, C.M.; Townsend, E.C.; Wingate, D.; Killpack, M.D. Learning nonlinear dynamic models of soft robots for model predictive control with neural networks. *Proc. IEEE Int. Conf. Soft Robot.* **2018**, *7*, 568–581.
128. Seel, K.; Grotli, E.I.; Moe, S.; Gravdahl, J.T.; Pettersen, K.Y. Neural network-based model predictive control with input-to-state stability. *Proc. Am. Control Conf.* **2021**, *5*, 3556–3563.
129. Ostafew, C.J.; Schoellig, A.P.; Barfoot, T.D. Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments. *Proc. IEEE Int. Conf. Robot. Autom.* **2014**, *5*, 4029–4036.
130. McKinnon, C.D.; Schoellig, A.P. Learning probabilistic models for safe predictive control in unknown environments. *Proc. IEEE Int. Conf. Robot. Autom.* **2019**, *6*, 2472–2479.
131. Hose, H.; Gräfe, A.; Trimpe, S. Parameter-adaptive approximate MPC: Tuning neural-network controllers without retraining. *Annu. Learn. Dyn. Control. Conf.* **2024**, *242*, 349–360.
132. Gros, S.; Zanon, M. Data-driven economic NMPC using reinforcement learning. *IEEE Trans. Autom. Control* **2020**, *2*, 636–648. [CrossRef]
133. Karg, B.; Lucia, S. Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Trans. Cybern.* **2020**, *9*, 3866–3878. [CrossRef]
134. Lovelett, R.J.; Dietrich, F.; Lee, S.; Kevrekidis, I.G. Some manifold learning considerations towards explicit model predictive control. *AIChE J.* **2019**, *66*, 16881. [CrossRef]
135. Klaučo, M.; Kalúz, M.; Kvasnica, M. Machine learning-based warm starting of active set methods in embedded model predictive control. *Eng. Appl. Artif. Intell.* **2019**, *1*, 1–8. [CrossRef]
136. Chen, S.W.; Wang, T.; Atanasov, N.; Kumar, V.; Morari, M. Large scale model predictive control with neural networks and primal active sets. *Automatica* **2022**, *1*, 109947. [CrossRef]
137. Chakrabarty, A.; Dinh, V.; Corless, M.J.; Rundell, A.E.; Zak, S.H.; Buzzard, G.T. Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences. *IEEE Trans. Autom. Control* **2017**, *1*, 135–148. [CrossRef]
138. Chen, S.; Saulnier, K.; Atanasov, N.; Lee, D.D.; Kumar, V.; Pappas, G.J.; Morari, M. Approximating explicit model predictive control using constrained neural networks. *Proc. Annu. Am. Control Conf.* **2018**, *6*, 1520–1527.
139. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.

140. Li, S.E. *Reinforcement Learning for Sequential Decision and Optimal Control*; Springer: Berlin/Heidelberg, Germany, 2023.
141. Bellman, R.E. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
142. Werbos, P. Consistency of HDP applied to a simple reinforcement learning problem. *Neural Netw.* **1990**, *3*, 179–189. [CrossRef]
143. Werbos, P. Approximate dynamic programming for real-time control and neural modeling. In *Handbook of Intelligent Control*; Van Nostrand Reinhold: New York, NY, USA, 1992.
144. Sutton, R.S. Learning to predict by the methods of temporal differences. *Mach. Learn.* **1988**, *3*, 9–44. [CrossRef]
145. Watkins, C. Learning from Delayed Rewards. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 1989.
146. Wang, Z.H.; Justin, S. Continuous-time stochastic gradient descent for optimizing over the stationary distribution of stochastic differential equations. *Math. Financ.* **2023**, *34*, 348–424. [CrossRef]
147. Moerl, T.M.; Broekens, J.; Plaat, A. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.* **2023**, *16*, 1–118.
148. Vamvoudakis, K.G.; Lewis, F.L. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica* **2010**, *46*, 878–888. [CrossRef]
149. Liu, D.; Wei, Q. Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems. *IEEE Trans. Cybern.* **2013**, *43*, 779–789. [PubMed]
150. Abu-Khalaf, M.; Lewis, F.L.; Huang, J. Neuro dynamic programming and zero-sum games for constrained control systems. *IEEE Trans. Neural Netw.* **2008**, *19*, 1243–1252. [CrossRef]
151. Feng, Y.; Anderson, B.D.O.; Rotkowitz, M. A game theoretic algorithm to compute local stabilizing solutions to HJBI equations in nonlinear H∞ control. *Automatica* **2009**, *45*, 881–888. [CrossRef]
152. Wang, D.; Liu, D.; Wei, Q. Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach. *Neurocomputing* **2012**, *78*, 14–22. [CrossRef]
153. Garaffa, L.C.; Basso, M.; Konzen, A.A.; de Freitas, E.P. Reinforcement Learning for Mobile Robotics Exploration: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3796–3810. [CrossRef]
154. Liu, D.; Zhang, H. A neural dynamic programming approach for learning control of failure avoidance problems. *Int. J. Intell. Control Syst.* **2005**, *10*, 21–32.
155. Si, J.; Wang, Y.T. Online learning control by association and reinforcement. *IEEE Trans. Neural Netw.* **2001**, *12*, 264–276. [CrossRef]
156. Long, M.; Su, H.; Zeng, Z. Output-feedback global consensus of discrete-time multiagent systems subject to input saturation via Q-Learning method. *IEEE Trans. Cybern.* **2022**, *52*, 1661–1670. [CrossRef]
157. Song, S.; Zhu, M.; Dai, X.; Gong, D. Model-free optimal tracking control of nonlinear input-affine discrete-time systems via an iterative deterministic Q-learning algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 999–1012. [CrossRef]
158. Liu, F.; Liu, Q.; Tao, Q. Deep reinforcement learning based energy storage management strategy considering prediction intervals of wind power. *Int. J. Electr. Power Energy Syst.* **2023**, *145*, 108608. [CrossRef]
159. Xu, X.; Hu, D.; Lu, X. Kernel-based least squares policy iteration for reinforcement learning. *IEEE Trans. Neural Netw.* **2007**, *18*, 973–992. [CrossRef] [PubMed]
160. Sartoretti, G.; Paivine, W.; Shi, Y.; Wu, Y.; Choset, H. Distributed learning of decentralized control policies for articulated mobile robots. *IEEE Trans. Robot.* **2019**, *35*, 1109–1122. [CrossRef]
161. Zhang, M.; Geng, X.; Bruce, J.; Caluwaerts, K.; Vespignani, M.; SunSpiral, V.; Abbeel, P.; Levine, S. Deep reinforcement learning for tensegrity robot locomotion. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 634–641.
162. Wang, X.; Deng, H.; Ye, X. Model-free nonlinear robust control design via online critic learning. *ISA Trans.* **2022**, *129*, 446–459. [CrossRef]
163. Xia, L.; Li, Q.; Song, R.; Modares, H. Optimal synchronization control of heterogeneous asymmetric input constrained unknown nonlinear MASs via reinforcement learning. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 520–532. [CrossRef]
164. Yong, H.; Seo, J.; Kim, J.; Kim, M.; Choi, J. Suspension control strategies using switched soft actor-critic models for real roads. *IEEE Trans. Ind. Electron.* **2023**, *70*, 824–832. [CrossRef]
165. Nassef, A.M.; Abdelkareem, M.A.; Maghrabie, H.M.; Baroutaji, A. Review of metaheuristic optimization algorithms for power systems problems. *Sustainability* **2023**, *15*, 9434. [CrossRef]
166. Sehgal, A.; Ward, N.; La H.; Louis, S. Automatic parameter optimization using genetic algorithm in deep reinforcement learning for robotic manipulation tasks. *arXiv* **2022**, arXiv:2204.03656.
167. Erodotou, P.; Voutsas, E.; Sarimveis, H. A genetic algorithm approach for parameter estimation in vapour-liquid thermodynamic modelling problems. *Comput. Chem. Eng.* **2020**, *134*, 106684. [CrossRef]
168. Qian, S.; Ye, Y.; Liu, Y.; Xu, G. An improved binary differential evolution algorithm for optimizing PWM control laws of power inverters. *Optim. Eng.* **2018**, *19*, 271–296. [CrossRef]
169. Godoy, R.B.; Pinto, J.O.; Canesin, C.A.; Coelho, E.A.; Pinto, A.M. Differential-evolution-based optimization of the dynamic response for parallel operation of inverters with no controller interconnection. *IEEE Trans. Ind. Electron.* **2011**, *59*, 2859–2866. [CrossRef]

170. Wang, D.; Sun, X.; Kang, H.; Shen, Y.; Chen, Q. Heterogeneous differential evolution algorithm for parameter estimation of solar photovoltaic models. *Energy Rep.* **2022**, *8*, 4724–4746. [CrossRef]
171. Shi, J.; Mi, Q.; Cao, W.; Zhou, L. Optimizing BLDC motor drive performance using particle swarm algorithm-tuned fuzzy logic controller. *SN Appl. Sci.* **2022**, *4*, 293. [CrossRef]
172. Hafez, I.; Dhaouadi, R. Parameter identification of DC motor drive systems using particle swarm optimization. In Proceedings of the 2021 International Conference on Engineering and Emerging Technologies, Istanbul, Turkey, 27–28 October 2021; pp. 1–6.
173. Tungadio, D.H.; Numbi, B.P.; Siti, M.W.; Jimoh, A.A. Particle swarm optimization for power system state estimation. *Neurocomputing* **2015**, *148*, 175–180. [CrossRef]
174. Foong, W.K.; Maier, H.R.; Simpson, A.R. Ant colony optimization for power plant maintenance scheduling optimization. In Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, Washington, DC, USA, 25–29 June 2005; pp. 249–256.
175. Lee, K.Y.; Vlachogiannis, J.G. Optimization of power systems based on ant colony system algorithms: An overview. In Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, Arlington, VA, USA, 6–10 November 2005; pp. 22–35.
176. Sitarz, P.; Powałka, B. Modal parameters estimation using ant colony optimisation algorithm. *Mech. Syst. Signal Process.* **2016**, *76*, 531–554. [CrossRef]
177. Zao, R.Z. Simulated annealing based multi-object optimal planning of passive power filters. In Proceedings of the 2005 IEEE/PES Transmission and Distribution Conference and Exposition: Asia and Pacific, Dalian, China, 18 August 2005; pp. 1–5.

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.