



Combining neural networks and control: potentialities, patterns and perspectives

Sophie Cerf, Eric Rutten

► To cite this version:

Sophie Cerf, Eric Rutten. Combining neural networks and control: potentialities, patterns and perspectives. IFAC 2023 - 22nd World Congress of the International Federation of Automatic Control, International Federation of Automatic Control, Jul 2023, Yokohama, Japan. hal-04060379

HAL Id: hal-04060379

<https://inria.hal.science/hal-04060379>

Submitted on 6 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Combining neural networks and control: potentialities, patterns and perspectives^{*}

Sophie Cerf^{*} Éric Rutten^{**}

^{*} Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRISTAL,
F-59000 Lille, France (e-mail: sophie.cerf@inria.fr)

^{**} Univ. Grenoble Alpes, Inria, CNRS, LIG, Grenoble, F-38000,
France (e-mail: eric.rutten@inria.fr)

Abstract: Machine learning tools are widely used for knowledge extraction, modeling, and decision tasks; a range of problems that Control Theory also tackles. Their relations have been largely explored by looking at stochastic control and Markov Decision Processes, due to the proximity of their formulations. However, novel links between machine learning and deterministic control are emerging; combining both approaches, e.g. by performing identification with learning, or controlling the training process. The recent flourishing literature is vast: there is a need to identify challenges, trends and opportunities on this interface. This survey contributes i) to the compared analysis of both fields. ii) Based on literature review, a categorization of combinations of learning and control is drawn. In the control framework, learning has been used for modeling, controllers tuning or adaptation, generating a controller or as a controller itself, for translating complex objectives, or checking controlled systems. Conversely, in the learning framework, control is used for tuning hyperparameters, selecting or generating training data, as the training or decision-making algorithm itself or to guarantee learning properties. iii) Finally, discussions on the literature open novel promising combinations to be explored, such as control of neural networks' training process.

Keywords: Machine Learning, Neural Networks, Learning for control, Reinforcement learning and deep learning in control, Machine learning in modelling, prediction, control and automation, Data-driven control

1. INTRODUCTION

Machine learning has emerged as a technology at the heart of present society's, particularly for tasks related to modeling and decision-making. Control theory is the well established field dedicated to the study and control of dynamical systems. Machine Learning (ML) and Control Theory (CT) are both used for a very wide range of problems, e.g., modelling and system identification, model reduction, control and decision-making, performance optimization, fault detection and isolation, and state and parameter estimation. They share a common interest in given scientific challenges, even a resemblance in their formulation, and most of all a complementarity in their data vs. model approaches. Works lying at the edge of learning and control theories can be found since the early 90s. The branch of control theory considering stochastic systems such as Markov Decision Processes (MDPs) is, historically and by problem statement, broadly linked with machine learning, and in particular its Reinforcement Learning (RL) formulation. Interactions and combinations of MDPs and RL are well documented (Van Otterlo and Wiering, 2012). Similarly, the link between fuzzy logic and control theory is well established in both communities (Lee, 1990).

Recent works combining control and learning explore novel directions, such as the connections that have been made between ML statistical analysis tools and model identification techniques (Matni et al., 2019; Morimoto and Doya, 2005), or the use of control-based algorithms for ML optimization (Kolarijani et al., 2018; Zanon and Gros, 2019). These novel contributions show promising interdisciplinary research directions. With this survey, we show that works combining control and learning can take a very wide variety of forms; from control of ML learning phase to learning of control cost functions. By gathering representative works among all this variety, we advocate that learning and control combinations should be considered in their variety, allowing to picture open and promising research directions. Despite the flourishing works at the intersection of both fields, much more developments on the topic are expected to arise in the next decade. Works providing a comprehensive and structured vision are urgently needed.

We set the scope of this survey solely to deterministic aspects of control theory. The case when random noise with known probability distribution affects the evolution and observation of the state variables will not be discussed here, readers interested in this topic can refer to the already existing literature (Van Otterlo and Wiering, 2012; Bensoussan et al., 2020). From the learning field perspective, this survey focuses on combinations considering neural network-based machine learning techniques,

^{*} The authors acknowledge the support of a research action in the Inria/Nokia-Bell Labs joint lab.

as it emerged as the most promising ML form and is now widespread on many applications (Rojas, 2013). Earlier survey works tackle the use of CT for other ML techniques (Lee, 1990; Dracopoulos, 1997).

This paper focuses on the combinations of non-stochastic control theory and neural-based learning, in the many forms they can take. We motivate those combinations by a deep analysis of each field. When put in perspective, their benefits and drawbacks show promising perspectives regarding their complementary. On the one hand, learning can be very accurate on large scale complex systems but is missing safety guarantees and is very dependent on the training data quality. On the other hand, control gives a strong theoretical background for proofs of stability, performance, and robustness. Still, it has constraints on the underlying system’s dynamics and objectives to reach, and does not scale easily in the number of considered state variables. An extended comparative analysis of both fields is still missing, it would enable motivating the control and learning combination concept formally.

Recent works can be found that enable both communities to communicate (Recht, 2019; Buşoniu et al., 2018; Bensoussan et al., 2020), describing the ML and CT combination as actionable intelligence or learning control (Mishra et al., 2018). (Moe et al., 2018) overview of the use of ML in a control framework from the robotic perspective. Reviews on specific combinations can be found, such as system identification and kernel-based learning (Chiuso and Pillonetto, 2019), reinforcement learning and self-tuning control (Matni et al., 2019), or reinforcement learning and optimal control (Kiumarsi et al., 2017). A comprehensive survey gathering and classifying the research works to highlight their diversity is nevertheless missing. Additionally, while numerous works combining CT and ML can be found, a classification schema that could position all those contributions in the big picture of the ML and CT combination is still missing. Such classification will help to bring out the promising research directions yet only emerging, and initiate ideas of new combinations.

Contributions of this paper are fourfold:

- (1) Comparative analysis of non-stochastic control theory and neural-based machine learning (based on strong points and limitations), further supporting the motivation of the existing works.
- (2) Gathering of works at the intersection of both fields.
- (3) Categorization of state of the art based on the type of control and learning combination.
- (4) Discussion upon the distribution of works among the various categories, and suggestions of unexplored combination research directions.

The categorization—the main contribution of the survey—is presented in Sections 4 and 5 as a result of the analysis of the existing state-of-the-art and envisioned future work, and illustrated in Fig. 1. A first distinction is made between works with a control frameworks enhanced by ML tools, and works with a learning framework enhanced by CT elements. Our study shows that, in a control framework, ML has been used for modeling, controllers tuning or adaptation, generating a controller or as a controller itself, translating complex objectives, or checking controlled systems. Conversely, in a learning formulation, one can

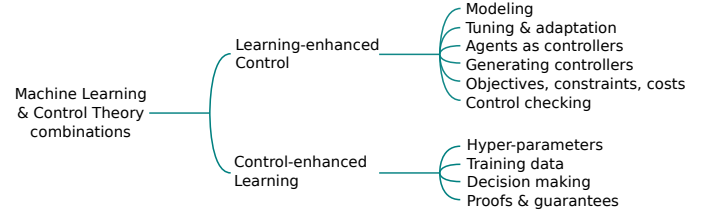


Fig. 1. Categorization of neural-based machine learning and non-stochastic control combinations, following the outlines of the literature review sections (Sections 4 and 5).

find control used for tuning hyperparameters, selecting or generating training data, as the decision-making algorithm itself or to guarantee learning properties.

The remaining of this paper is organized as follows. After a brief introduction on machine learning and control theory in Section 2, an analysis of both fields’ advantages and limitations are given in Section 3, later put in perspective to motivate their combination. Section 4 presents and categorize the literature on learning-enhanced control; and Section 5 focuses on control-enhanced learning. Eventually, perspectives on established, emerging and open research directions are given (Section 7).

2. BACKGROUND ON LEARNING AND CONTROL

Control and learning theories are now presented, giving the important notions used in the remaining of the paper.

2.1 Machine Learning and Neural Networks

Machine learning extracts statistical knowledge from data to make predictions or to take actions (Mohri et al., 2018). Neural Networks (Rojas, 2013) are models used to represent this knowledge that leverages features in a prediction through several layers of unit functions called neurons (or perceptron). The key notions articulated in neural networks and machine learning are the following: *i)* raw data, organized in samples, without constraints on their form (it can be images, time-series, etc.) but often described using features, *ii)* a task, among classification, regression, and decision-making on actions, *iii)* a model form and configuration (such as NN layers nature and size) and finally *iv)* an optimization method, that adapts the model parameters to the data during the training phase. Validation of the network form is experimentally done with a validation dataset, and the overall solution is then evaluated using a test set. A distinction can be made between (un)supervised learning tasks consisting of classification and prediction done offline, strictly after the data collection time; and reinforcement learning (RL), where decisions of actions are taken in an online interactive fashion (Alpaydin, 2020). The need for data labels also characterize classical learning in comparison with RL, where a reward function is needed. Despite the very broad forms learning can take, this survey focuses on neural networks, as the most prevalent and performing learning technique (Rojas, 2013).

(Un)supervised Learning Illustration of schematic concepts is given in Fig. 2a. One can also refer to classic textbooks (Mohri et al., 2018; Alpaydin, 2020). In

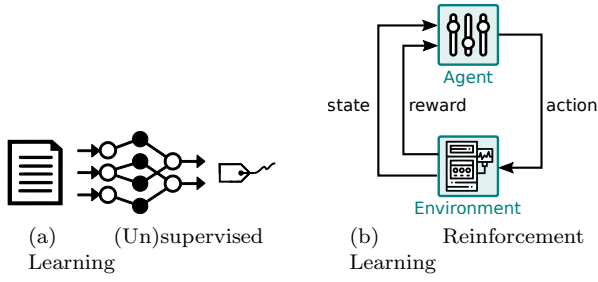


Fig. 2. Different Machine Learning Setups.

(un)supervised learning, the task consists in predicting a label for each data instance. This label can be either a discrete class (classification task) or a continuous value (regression task). To realize those tasks, a machine learning model is used that can take many forms and sizes. Common examples are k-nearest-neighbors, decision trees, support vector machines or neural networks. Decision has to be made on which model to use and with which configuration or hyperparameters, such as the neural network number of layers, the connectivity pattern, etc. Those choices are most of the time done through exploratory search of the configuration set, with limitation regarding combinatorial explosion.

Then, a model optimization algorithm, such as gradient descent, is used to fit the model parameters to the data. This is the training phase. It is important to distinguish the model's parameters (such as the weights and biases of a neural network) and hyperparameters (e.g., number of neuron per layer) as the first are mathematically optimally found—mainly using stochastic gradient descent—and the other result of tuning decisions—as trial and error with cross-validation. Training is often realized using a learning rate factor, a hyperparameter reflecting how much a new sample will modify the current model. In supervised learning, training data instances are considered with their corresponding label and the optimizer minimizes the prediction error of the set, with some error margin allowed for generalization. In unsupervised learning, no label is available, the task consists then in generating classes that maximize inner similarity and outer differences.

Once trained, the resulting machine learning model is evaluated using a test set. Data labels are predicted and compared to ground truth using metrics such as accuracy, precision, and recall (either per class or aggregated), or confusion matrices. A validation set can also be used to decide upon various algorithm configurations of hyperparameters. The trained and tested algorithm is then ready to be used to realize its main task.

Reinforcement Learning In a reinforcement learning scenario, the learning algorithm, called agent, interacts with its environment in a reactive fashion, see Fig. 2b. Dedicated textbooks (Sutton and Barto, 2018) and classes (Silver, 2015) on the topic can be found. The main difference with (un)supervised learning are the algorithm outputting an action to take on the environment, receiving a reward about the former actions, and perceiving the environment states. All this is repeated cyclically, in a form of feedback loop. Both learning and usage of the algorithm is done at each time-step. The reinforcement learning problem

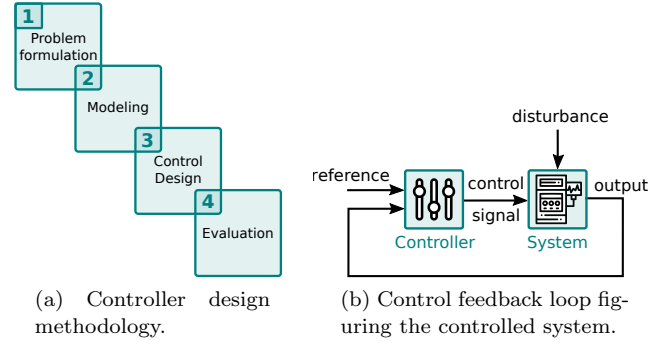


Fig. 3. Control Theory: methodology and feedback loop.

consists of finding the best actions so that to maximize reward. A policy is defined as an algorithm mapping the states to the actions, and the value function as giving the reward based on the states. It decouples the prediction problem from the control one.

There is a broad variety of methods for solving a reinforcement learning problem. Among the most used are Q-learning, that finds the optimal policy by maximizing the expected value of the total reward, and Actor-Critic, where both policy and value function are learned, with the estimated value function criticizing on policy's actions. Both model-based and model-free approaches exist. When they are used, models can take many forms, for instance Markov transition matrices or neural networks.

As value function and policy search are simultaneously done on the same data, there is an inherent trade-off between state-space exploration, trying out new actions to increase knowledge about the environment, risking reaching undesirable states; and exploitation, in which the best known policy is applied, knowing that there may be a better unexplored solution.

2.2 Control Theory

Overview Control theory enables to monitor a dynamic system and provides *deterministic* guarantees that its state will behave according to objectives and reach a predefined desired state. Depending on the properties of the system, different control theories exist. For systems with signals having continuous values, continuous control is used. When a system's states take discrete, logical values, Discrete Event Systems (DES) theory is used with tools such as Petri nets or Finite State Machines. Developing control is a fourfold process, represented in Fig. 3a: *i*) problem formulation, including at least one control knob and one observable state and the definition of objectives (among stability, reference tracking, robustness, disturbance rejection, and extremum seeking); *ii*) mathematical modeling, either from already known behavior equations or by identification from data; *iii*) choice of the controller type (PID, H_∞ , optimal, MPC, etc.) and its parametrization, *iv*) analytical evaluation providing guarantees.

Control of Deterministic Continuous Dynamic Systems

Continuous control realizes dynamic systems' configuration, to bring them to a desired state through the use of feedback loops. Refer to Fig. 3b for an illustration, (Doyle et al., 2013) for a classic textbook and (Hellerstein

et al., 2004), a book dedicated to control application to computing systems. Such control problem formulation considers a system evolving through time, tunable by at least one signal (called control signal, it is the controllability condition) and observable by at least another signal (called state, it is the observability condition). The system can also be under the influence of an uncontrollable signal (called disturbance). Both steady state (stabilized system) and transient (evolving through time) behaviors are considered in continuous control. Controllers can meet five objectives: (i) stability, ensuring that the system’s state will not diverge; (ii) reference tracking, i.e., bringing the state to an objective reference value or signal, with both steady state and transient behavior objectives; (iii) robustness to measurement noise or system’s variations; (iv) disturbance rejection, guaranteeing that the disturbance will not impact (too much) the system; and (v) extremum seeking, to drive system to some form of optimality.

A control setup consists in an algorithm deciding on the control signal and a system to which it is applied. Many types of algorithms, i.e., controllers, exist such as Proportional Integral Derivative, feedforward, optimal, Model Predictive Controller (MPC), or robust (H_∞) controllers. They differ in their hypothesis regarding the system, the information they require, on their computing complexity, and on the guarantees they provide. It is up to the control engineer to decide which controller to use.

Controllers’ design, consisting in finding adequate controller’s parametrization, is an important step as it will enable to derive guarantees on the closed loop behavior. Their configuration is often based on a model of the system under study, found either by mathematical analysis or by black-box identification experiments on the system in case of data-driven control. The model is most of the time linear (between the control signal and the state) even when the underlying system is not, as the model does not have to be perfectly accurate but rather adequate to optimize the controller design. More complex non-linear models can be used if necessary, however it induces extra computing. Controller design often requires full knowledge of the system’s states, as provided by so-called observer or estimator algorithms. Adaptation technique enables self-tuning of controllers at runtime. It consists in using optimization algorithms to update in real-time the controller’s parameters to deal with system uncertainties/non-linearities or unmeasured external disturbances.

Mathematical analysis of the controlled system enables to guarantee that the objectives will be satisfied (Khalil and Grizzle, 2002). When realized experimentally and systematically, this step is called control checking. Preliminary to the control conception, some specific analysis tools can be used to evaluate to what extent the system is controllable and observable, and thus deciding if (and what kind of) control can be applied.

3. COMPARATIVE ANALYSIS

Neural-based learning and deterministic control tackle a similar range of problems, such as modeling and decision-making on dynamical systems. This section sets the strength and limitations of each approach and concludes with a comparative analysis, showing complementarities

Table 1. NN/RL-based learning strengths and limitations.

Strengths		Limitations	
L^+1	Performance	L^-1	Stability
L^+2	Complex applications	L^-2	Accuracy guarantees
L^+3	Scalable	L^-3	Data richness
L^+4	Generalization	L^-4	Self-adaptation
L^+5	Model-free	L^-5	Hyper-parameters
L^+6	Experimental Evaluation	L^-6	Efficiency
		L^-7	Test dataset
		L^-8	Explainability

that further motivate their combinations in the many forms they can take.

3.1 Neural Networks

Strengths Performance is NN approaches’ strongest asset (Aggarwal et al., 2018) (L^+1). Learning techniques often outperform concurrent approaches for a given problem regarding the accuracy aspect. Learning can be applied to various contexts and types of data (Abiodun et al., 2018) with complex non-linear behaviors (Stengel, 1991) (L^+2) and scales to very high dimensional problems (Stengel, 1991; Behler, 2015) (L^+3). Generalization and transfer learning make the algorithms reusable in various contexts (Novak et al., 2018) (L^+4). In model-free approaches, no initial knowledge about the features’ interaction or environment behavior is needed (L^+5). Learning is hardly sensitive to biases introduced by modeling errors (Matni et al., 2019). Evaluation using extensive datasets shows the efficiency and practical usage of algorithms, even under adversarial conditions (Larochelle et al., 2007) (L^+6).

Limitations While task performance reaches very good accuracy *on average*, few stability guarantees are given for worst-case scenarios (L^-1). It is common to consider that reinforcement learning is “required to fail to learn”, i.e., adopt non-desirable behavior (Sutton and Barto, 2018). Safety of learning agent decision is thus a real challenge. Algorithms lack strong theoretical deterministic guarantees regarding the convergence of predictions or actions to ground truth or desired values (L^-2), even if effort is put to find proof once the network is designed (Matni et al., 2019). Moreover, the algorithm’s performance is highly dependent on the learning data richness in (un)supervised learning, and small perturbations can cause network failure (Szegedy et al., 2013) (L^-3). Trained algorithms are not, by default, designed to be adaptable to changes in the environment behavior law or features interactions as in the concept of life-long learning (Parisi et al., 2019) (L^-4), it is mostly achieved by a side effect of generalization. Finding optimal values for networks architecture and hyperparameters (depth, learning rate, exploration, etc.) is either very costly or requires the use of experts (Liu et al., 2021; Ren, 2021) (L^-5). While the training phase can be really long and requires numerous samples, most of the information is learned in the first epochs, leading to poor accuracy gain in the last epochs, i.e., poor sample and energy efficiency (Aggarwal et al., 2018; Sze et al., 2017) (L^-6). Evaluation of algorithms’ performance on a real system is not always straightforward, as it relies on the hypothesis that an appropriate test dataset is available (Aggarwal et al., 2018; Martin et al., 2021) (L^-7).

Table 2. Control strengths and limitations.

Strengths		Limitations	
C ⁺ 1	Dynamic Perspective	C ⁻ 1	Dynamical systems
C ⁺ 2	Problems decoupling	C ⁻ 2	Signal-based systems
C ⁺ 3	Feedback principle	C ⁻ 3	Not scaling
C ⁺ 4	Worst-case guarantees	C ⁻ 4	Limited systems complexity
C ⁺ 5	Robustness	C ⁻ 5	Model-based
C ⁺ 6	Safety	C ⁻ 6	Limited objectives
C ⁺ 7	Trade-offs	C ⁻ 7	Computational cost
C ⁺ 8	Analysis tools	C ⁻ 8	Sub-optimal performance
C ⁺ 9	Model-based	C ⁻ 9	Experimental Evaluation
C ⁺ 10	Explainability		

Traceability of algorithms’ decisions and interpretability or explainability is so far limited (Xu et al., 2019) (L⁻8).

3.2 Non-stochastic Control

Strengths Control Theory gives special attention to the time dynamic perspective when considering a system (Ljung and Glad, 1994) (C⁺1). The tasks of environment and state observation and estimation can be decoupled from the decision one (Besançon, 2007) (C⁺2). The feedback principle—taking into account current state in the decision process—is a very useful paradigm, allowing for simplicity in algorithms and adaptation to the environment (Doyle et al., 2013) (C⁺3). Control mathematical formulation enables to derive worst-case deterministic guarantees both on the system stability and on its performance in a closed-loop (Hinrichsen and Pritchard, 2005) (C⁺4). Further theoretical guarantees of robustness and adaptation to the system and environment changes, uncertainties, and faults can be retrieved (Landau et al., 2011; Amato, 2006) (C⁺5). Control guarantees safe behaviors (Matni et al., 2019), and DES controllers are correct by construction (C⁺6). Performance can explicitly be traded-off with robustness and cost-efficiency (Kirk, 2004) (C⁺7). Analysis tools allow for theoretic determination of the controllability and observability of a system, thus defining the set of reachable values of the objectives (Hespanha, 2018) (C⁺8). The control theory approach is knowledge-based, thus avoiding a large preliminary exploration phase, a better sample efficiency and ensuring performance from the first instant (Doyle et al., 2013; Hou and Wang, 2013) (C⁺9). By construction, decisions taken are traceable, explainable, and interpretable (Sawant and Gros, 2022) (C⁺10).

Limitations The control framework addresses only dynamical (C⁻1), signal-based systems (C⁻2) (Franklin et al., 2002), with a limited number of variables in practice (Liu et al., 2019) (C⁻3). Dealing with non-linearities requires expert-knowledge and remains a flourishing research topic (Mayne, 2000) (C⁻4). Initial knowledge about the system is commonly required, such as equations of its dynamic behavior (Brosilow and Joseph, 2002) (C⁻5). The control framework is limited to specific classes of objectives such as stability, tracking, extremum seeking and robustness (Doyle et al., 2013) (C⁻6). Synthesis algorithms generating DES controllers suffer from combinatorial explosions (Wonham, 2015); while advanced control technique lack practicality in their real-time implementation (Alamir, 2020) (C⁻7). Robust and optimal designs are hardly concealable (Doyle, 1978), possibly resulting in sub-

Table 3. Learning-enhanced control strengths
Colors refer to **established**, **emerging** and novel
research directions (refer to Section 6).

		Learning strengths					
		Performance	Complex applications	Scalable	Generalization	Model-free	Experimental Evaluation
Control limitations		L+1	L+2	L+3	L+4	L+5	L+6
Dynamical systems	C ⁻ 1		RD3				
Signal-based systems	C ⁻ 2		RD3				
Not scaling	C ⁻ 3			RD4			
Limited complexity	C ⁻ 4		RD3				
Model-based	C ⁻ 5	RD1			RD6	RD6	
Limited objectives	C ⁻ 6		RD5				
Computational cost	C ⁻ 7			RD4			
Suboptimal performance	C ⁻ 8	RD2					
Experimental Evaluation	C ⁻ 9						RD7

optimal performances (C⁻8). As stability and performance are mathematically guaranteed, control applications are sometimes missing a comprehensive and reproducible evaluation on a challenging real-world setup (C⁻9), or in a finite-time scenario (Matni et al., 2019).

3.3 Complementary weaknesses and strengths

When put in perspective, the strengths and limitations of both theories turn out to be largely complementary. Additionally, to the motivation brought to light for combining them, point to point comparison also brings out research directions. First, control’s drawbacks are mitigated thanks to learning, see Table 3 for a synthetic view. Then we envisage learning’s limitations through control’s strong points, with research directions summarized in Table 4.

Learning to foster control Putting first in perspective ML strengths and CT limitations, several promising combinations appear. NN high prediction performance can help build more accurate control systems in the model construction (RD1) and controller design (RD2). ML’s ability to deal with complex, highly non-linear (RD3) and high dimensional (RD4) setups could enable extending control theory to a larger class of systems. Objectives CT can fulfill are restricted, using RL-inspired formulation could broaden CT spectrum (e.g., queries answering, image processing, or transfer tasks (RD5). Moreover, the ML model-free approach and generalization capacities could benefit to control, for instance by loosening the requirements for using adaptation tools (RD6). Eventually, ML’s extensive and reproducible evaluation methodology on large sets of real data can help consolidate control results on practical systems (RD7).

Control to foster learning Considering now the complementary point of view of using control to enhance learning, one can first think of using the concept of feedback to monitor the hyperparameters’ configuration (RD8) or training data selection (RD9) based on learning efficiency. The guarantees of stability of controllers can help build safe ML (RD10). Performance guarantees provided by controllers can be transposed to learning systems to guarantee

Table 4. Control-based learning roots of success.
Colors refer to *established*, *emerging* and novel research directions (refer to Section 6).

Learning limitations	Control strengths										
	Dynamic Perspective Problems decoupling Feedback principle Worst-case guarantees	C+1	C+2	C+3	C+4	C+5	C+6	C+7	C+8	C+9	C+10
Stability	L ⁻ 1				RD10		RD10				
Accuracy guarantees	L ⁻ 2		RD12		RD11				RD12		
Data richness	L ⁻ 3					RD16				RD14	
Self-adaptation	L ⁻ 4					RD15					
Hyperparameters	L ⁻ 5	RD8		RD8							
Efficiency	L ⁻ 6			RD9						RD14	
Test dataset	L ⁻ 7								RD13		
Explainability	L ⁻ 8										RD17

algorithms’ accuracy (RD11). Control analysis tools of observability, controllability and can help refine ML working set and reachable objectives (RD12). ML algorithms theoretical validation, complementary to evaluation with test sets, can be inspired by similar deterministic analysis techniques as CT (RD13). CT model-based approach can help build ML algorithms that efficiently work from the initial instants, thus reducing exploration and, therefore, the training time of ML (RD14). CT adaptation tools can help ML algorithms to learn continuously changing systems (RD15). ML algorithms’ robustness can be improved using CT techniques to cope with model uncertainties, for instance, when too few/poor data are available (RD16). Eventually, the traceability of the CT decision could help improve ML’s explainability (RD17).

Numerous potential combinations arise, naturally divided according to which theory builds on the classic formulation of the other to overcome its limitations. The next sections review existing work in those two families, further identifying specific patterns. Section 6 links the state of the art to the motivation analysis, revealing perspectives of emerging and novel open research directions.

4. LEARNING-ENHANCED CONTROL

In this section, we only consider works where the classic control framework is augmented using neural networks. The reverse setup—a learning framework supported with control—is presented in Section 5.

Benefits and limitations. The main motivation for such combinations is improving the performance of the controlled system (RD2) and extend theory towards better handling non-linear systems. However, precautions have to be taken to keep the stability and convergence guarantees. Improvement of this guarantees-performance trade-off is at the cost of constraints on the system’s size and signal properties, derived from the control formulation.

Composition patterns. Two major approaches are identified: i) a learning algorithm used to derive a model followed by a classic controller design phase, ii) a controller tuned or adapted with ML. Fewer works use a learning algorithm directly as a controller, to generate an appropriate one, to translate complex control objectives, and to check

the control. All combinations are detailed in this section, and schematically represented in Fig. 4.

4.1 Learning for modeling

Identification Neural Networks can learn a control model offline, from data, improving accuracy and dealing with complex non-linearities. The stable neuro-controller (Kretschmar, 2000) learns a model with Reinforcement Learning for the design of a robust controller. Stability is ensured, even during training, and worst possible guarantees are given for input disturbances (Morimoto and Doya, 2005). Strict worst-case guarantees can be relaxed to robustness with high probability (Matni et al., 2019). Finite time guarantees, rather than usual asymptotic ones, were recently proved for different types of systems: marginally stable (Simchowitz et al., 2018), unstable (Sarkar and Rakhlin, 2018), partially observed (Sarkar et al., 2019) and sparse systems (Fattahi et al., 2019). Advanced ML structures are also used for identification, such as Recurrent NN (Wu et al., 2019), physics informed NN (Barreau et al., 2021) and kernel-based learning (Chiuso and Pilonetto, 2019), particularly promising for non-linear systems. Learning based identification also applies to hybrid systems—combining continuous fluid dynamics with discrete jumps (Lauer, 2019; Makdesi et al., 2021).

Orthogonally, machine learning can be used in analytical modeling approaches, i.e., first principles models. Neural networks can approximate ordinary and partial differential equations ((Lagaris et al., 1998)). Recent works of (Sirignano and Spiliopoulos, 2018) added proofs of convergence. (Chen et al., 2018a) introduce Neural ODE, a deep NN using black-box ODE solvers. Its main advantage lies in the explicit trade-off between computation speed and numerical precision. Neural ODEs have been attracting significant attention, e.g., for robustness (Yan et al., 2019) or for system identification (Du et al., 2020).

ML is also combined with Discrete Event Control. Identification of discrete models can involve optimization techniques (Basile et al., 2020), however, not in the scope of neural networks. Those works are guarantees-oriented.

Observation Observation problems rely on prediction, a well studied problem in learning. Data-driven meth-

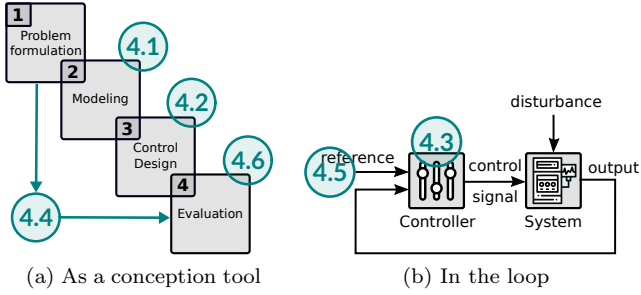


Fig. 4. Categorization of neural-based learning used in a control framework, with reference to sections detailing the state of the art.

ods foster output prediction when considering non-linear systems (Janny et al., 2021). In robotics applications, observation problems benefit from learning in an approach that is called *control from pixels*. Typically, a video stream is used as a sensor for the main control task. State estimation is performed using ML and combined with controllers in various contexts: for UAV flight control (Tang et al., 2018), LQR with nonlinear observations (Mhammedi et al., 2020), or for reference tracking problem in an optimal control formulation (Dean and Recht, 2021). The two latest works provide sample complexity and stability guarantees.

Online identification Online identification can also benefit from ML. (Rajamäki and Hämmäläinen, 2017) use a double learner approach based on decisions trees and neural networks, respectively, for fast but imprecise and accurate but slow modeling. Afterward, the model is fed to an MPC controller. (Van De Ven et al., 2007) combines several neural networks to model separately different parts of a control system. Such setup reduces the complexity of the models to learn, and ease the incorporation of external knowledge. A feedforward controller is then computed based on the identified models. Model-Free Control also benefits from learning (though not neural-based) to estimate the ultra-local model in real-time. This model is then used to tune an intelligent Proportional controller (Fliess and Join, 2020). Eventually, note that online observers (of states, and external disturbances) can also benefit from neural networks (Dierks and Jagannathan, 2009; Samy et al., 2010).

The online approaches however do not provide guarantees that controllers will be stable or accurate, as the optimization criteria is related to the modeling or the estimation error. This well known limitation in control theory has led to the theory of adaptive control. Note the work of (Hoffmann, 2015) that uses a learning algorithm for online modeling (in which inaccuracy regarding actual system’s performance are used to set the pole of the controller) with mathematical guarantees of stability.

4.2 Learning for tuning and adaptation

Another approach consists in using neural networks to learn the controller’s parameters, without going through a modeling phase. The learning of control parameters either requires offline training or can be performed online.

Tuning In the offline scenario, neural networks are trained to tune PID controllers. Focus is made on math-

ematical guarantees of stability and performance improvements—in comparison to contemporary numerical or graphical methods (Chu and Teng, 1999)—and on robustness (Lee et al., 2002). Automatic tuning of Linear Quadratic Regulators has also been studied using iterative learning (Fiechter, 1997; Dean et al., 2020a).

Adaptation Adaptation using learning started with neural networks adapting a controller’s parameters, for instance using the concept of back-propagation, generalizing the concept of memory with storage in the network weights (Guez et al., 1988; Miller et al., 1990). A link is made to classic control adaptation theory in the case when some nodes include the model parameters, see (Pioveso and Owens, 1991). Nowadays, flourishing works can be found, from curve fitting used to adapt a PI controller (Caldas et al., 2020) to RL adapting a PID (Neftci and Averbeck, 2019), a fuzzy controller (Jamshidi et al., 2016) or a non-linear controller (Banerjee and Chatterjee, 2019). Recent works have focused on deriving mathematical stability guarantees, using neural networks for adaptation (Lama and Zhou, 2013; Balakrishnan et al., 2008), or reinforcement learning for online identification and adaptation in approximate optimal regulation (Kamalapurkar et al., 2016) and in Economic Non-linear MPC (Gros and Zanon, 2019). See (Poveda et al., 2019) for a survey and (Benosman et al., 2019) special issue. Learning in the adaptive control context induces analysis of finite-time guarantees (Rantzer, 2018; Fazel et al., 2018) and sample-efficiency (Tu and Recht, 2019), see (Matni et al., 2019) for a general discussion. Eventually, the problem of controlling an unknown linear dynamical system in the presence of (non-stochastic) adversarial perturbations (and adversarial convex loss functions) has been tackled (Hazan et al., 2020), and extended with bounds computation and costs analysis (Chen and Hazan, 2021). The notion of adversity inspired from ML setups extends the scope of control’s possibilities and led to the concept of adversarial control.

4.3 Learning agents as controllers

Neural networks can directly be used as controllers. Combinations of neural networks and model predictive control, gathered under the terminology *Neural Predictive Controller* (Zamarreño and Vega, 1999). In such setup, the system’s behavior is predicted by a neural network, and this prediction is used by the MPC. This approach is mainly used to deal with non-linear systems ((Zamarreño and Vega, 1999; Lazar and Pastravanu, 2002)). Choosing the network shape leverages the type of controller, e.g., a decentralized controller in (Åkesson and Toivonen, 2006). Guarantees can be ensured (Korda, 2021), e.g., using constrained neural networks (Chen et al., 2018b).

An orthogonal approach keeps the control feedback-based schema without using models and classical controllers, but rather train a neural network to act as a controller. (Cong and Liang, 2009) use a NN as a non-linear adaptive controller, and provide guarantees of stability and performances. Advanced ML structures are also used, such as sigmoid-based NN (Ivanov et al., 2019) or reinforcement learning (Vrabie et al., 2009), both proved stable.

Different class of systems also benefit from NN controllers. Discrete-time systems can be controlled with NN, specially

those presenting non-linearities (Sarangapani, 2006; Gilra and Gerstner, 2018), e.g., for reference tracking (Pauli et al., 2021). Hierarchical control can be controlled with NN (Radac and Precup, 2016). Differently, from the DES perspective, (Yamasaki and Ushio, 2004) propose to use reinforcement learning to obtain a decentralized supervisor that coordinates existing local controllers.

4.4 Learning to generate controllers

Discrete Event Controllers can be automatically generated, i.e., computed *on-the-fly*. Doing so by using learning tools is useful when accurate descriptions of both the specifications and the DES are not available. The Reinforcement Learning setup is particularly suited to synthesize DES controllers (Dadone et al., 1998; Ushio and Yamasaki, 2003; Zhang et al., 2018). However, this approach does not strictly follow the methods of Supervisory Control, and does not target the same guarantees.

In continuous control, the approach called Machine Learning Control proposed by (Duriez et al., 2017) generates controllers and selects the best suitable one automatically. The underlying learning relies on genetic programming. To the best of the authors' knowledge, no equivalent setup exists using neural networks.

4.5 Learning objectives, constraints, and costs

Controllers in a tracking setup need to be fed with a reference signal. Getting a relevant reference value is not always straightforward, e.g., in an optimization problem or in a complex tracking setup. Conversely, neural networks can take as input fully raw data, such as images or video streams. Perception-based Control also called Feedback from Pixels investigates the use of learning to generate a control reference signal, mostly applied in the robotic context. (Jones and Soatto, 2011) generate a robot trajectory path from camera sensing, further sent to the controller. All other kinds of sensors can also be used (Lynen et al., 2013). (Abu-Khalaf et al., 2021) realizes tracking directly in image space. Robustness guarantees for such setups are provided by (Dean et al., 2020b).

In cloud applications regulation, ML is combined with a MAPE monitoring layer to generate set points for low-level controllers (Weyns et al., 2021). Using ML allows for environment estimation and adaptation to manage temporary infeasible objectives.

Learning-based model predictive control reformulates the MPC cost functions and constraints using learning with safety insurance (Hewing et al., 2020). This technique tackles the objectives' complexity using ML while the controller works in its classic formulation. Orthogonally, MPC real-time implementation can be fostered with ML: (Alamir, 2020) uses non-neural-based clustering algorithms to update stochastic cost and constraints to approximate a Stochastic Nonlinear MPC. The use of MPC for hybrid optimal control equally benefits from learning, e.g., by refining the state-space using Q-learning (Legat et al., 2021). The ML concept of *adaptive regret* shows promises in the field of control, particularly to tackle the time-varying linear dynamical systems (Gradu et al., 2020).

4.6 Learning for control checking

ML's methodology on extensive experimental evaluation inspires control community. (Yaghoubi and Fainekos, 2019) find examples of requirements breaks by feeding an optimal controller with neural networks' selected challenging inputs. (Ghai et al., 2021) is interested in generating maximally adversarial disturbances with online learning, and provides a solution able to adapt to control inputs. (Balasubramaniam, 2019) uses ML to represent controllers synthesized by model checkers, thus enabling experts to monitor and debug the system efficiently. Verification of control performance and determination of the region of attraction was performed using active learning (Tadiparthi and Bhattacharya, 2022a) or Physics-Informed Neural Networks (Tadiparthi and Bhattacharya, 2022b).

5. CONTROL-ENHANCED LEARNING

This section completes the literature survey with works that conversely use control elements in a learning setup. While the motivation analysis of Section 3.3 identifies numerous advantages—guarantees, robustness, safety—this perspective has gained notably less interest from both communities. **Benefits and limitations.** The prevalent motivation for this category of combinations is improving the learning process by fastening the training phase (RD9,14). Other motivations are increasing the learning robustness and its ability to learn continuously (RD15). Few works tackle stability and performance guarantees (RD10-12). The main limitation of those methods is the time and effort dedicated to the control design—and all the upstream phases of problem formulation and modeling—that may be negligible or prohibitive depending on the application.

Composition patterns. Combinations are divided in four categories: *i)* NN's training phase is seen as a dynamical system that can benefit from automation, e.g., by regulating its hyperparameters, *ii)* control techniques soundly select data to fasten learning when dealing with large training data, *iii)* decision-making in reinforcement learning is done using a control-based algorithm, and *iv)* stability and performance guarantees of neural learners are derived using control tools. All combinations are detailed in this section and illustrated in Fig. 5.

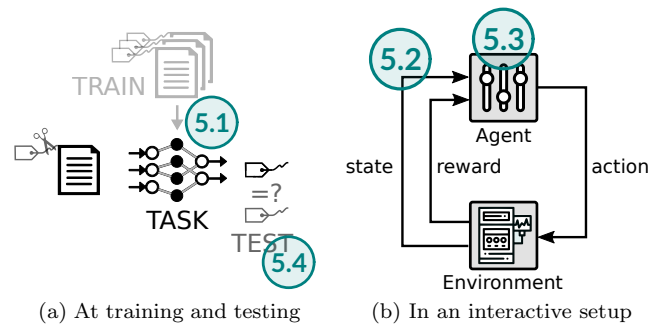


Fig. 5. Categorization of control elements used in a learning framework, with reference to sections detailing the state of the art.

5.1 Controlling hyper-parameters

The feedback principle—acting based on the system’s current state rather than with a blind predefined law—can improve learning performances. This idea is intuited by (Schaul et al., 2013) on the learning rate regulation, however, without a proper control formulation. (Zhao et al., 2020) control a convolutional neural network learning rate with the objective of minimizing the loss function. The motivation is to reduce training time without degrading accuracy, thanks to the use of event-based. Optimal control of both the learning rate and momentum hyperparameters shows competitive performance while being robust to changing model settings and datasets (Li et al., 2017).

The exploration to exploitation trade-off can be managed automatically, e.g., through the ϵ hyperparameter in ϵ -greedy exploration (Lowrey et al., 2018). (Tokic, 2010) proposes Value Difference Based Exploration (VDBE), an approach that dynamically computes a state-dependent exploration probability, increasing parameter robustness. However, to the best of our knowledge, there is no proper control formulation for regulation of exploration process.

5.2 Controlling training data

There exist a class of problems relying on decision-making that can be addressed both by learning and control techniques. The idea of training a network by copying an existing controller was intuited decades ago (Miller et al., 1990), motivated by scenarios where the learner would require more accessible measure signals.

In (Levine and Koltun, 2013) differential dynamic programming performs local trajectory optimization to generate a dataset further used to train a global policy through imitation learning. It incorporates control-based guiding samples into the policy search. In (Mordatch and Todorov, 2014) a control trajectory optimizer acts as a teacher and gradually guides the network towards better solutions. The controller finds solutions similar to the NN output, which makes the training easier. This approach combines the local optimality properties of control with the global policy search of RL. Advantage in using control also relies on its adaptation properties regarding how good the learning is performing.

In (Fröhlich et al., 2021), control solutions are used as initial policies to constrain the policy search space in Bayesian optimization. For problems where local optimization is sufficient, this solution increases sample efficiency and mitigate the curse of dimensionality.

5.3 Training and decision-making using Control

In model-based reinforcement learning, the optimal policy can be found using control theory. In this configuration, an MPC controller decides on an action to take on the agent, based on state info, reward, and an NN-based model of the system’s dynamics. Motivations of such combination are data-efficiency (Kamthe and Deisenroth, 2018; Gros and Zanon, 2020; Nagabandi et al., 2018), robustness to modeling uncertainties (Gros and Zanon, 2019) or to worst-case disturbances (Donti et al., 2020), or safety (Zanon and Gros, 2019; Gros et al., 2020). From

the control community perspective, controllers in a reinforcement learning framework realize complex objectives (Williams et al., 2017), i.e., more advanced than stability and reference tracking. Hybrid setup propose to jointly use model-based control and arbitrary differentiable policy as agent in a reinforcement learning setup (Zhang and Capel, 2021). The learning element provides universal function approximation, while the controller takes over around the operating point to stabilize the system.

Adaptive control schemes are used to find an optimal policy *online*, see (Khan et al., 2012) for a review. The validity of this setup extends to partially unknown systems (Moghadam and Lewis, 2019).

Orthogonally, optimal control can train a deep neural network as an alternative to the classic gradient-based search of the network optimal parameters (Li and Hao, 2018). (Kolarijani et al., 2018) use a control feedback approach is used to fasten the solving of gradient-based methods, additionally providing guarantees on the convergence rate. Regarding Neural ODE, their training in a final time horizon can be performed using optimal control (Esteve et al., 2020). Another perspective on Neural ODE considers training as a simultaneous control task (Ruiz-Balet and Zuazua, 2021), where a same common control should drive different initial datum to their associated label.

5.4 Proofs and guarantees using Control

Interest of the learning community on guaranteed behavior is reflected by the popularity of random neural networks, that are proved stable in steady-state (Gelenbe, 1993).

Control provides analysis tools of reachability set of data-driven methods, e.g., from noisy data without a priori model (Alanwar et al., 2021). Computing forward reachable sets allows for neural network verification. Control tools are used, considering neural network controllers as part of a closed-loop system in a model-based setup (Seidman et al., 2020; Weinan, 2017; Huang et al., 2019), for a neural network with a particular structure (Sidrane and Kochenderfer, 2019; Tran et al., 2020; Newton and Papachristodoulou, 2021), and more recently in a model-free approach (Thorpe et al., 2021) or on a partially observable system (Kirschner and Krause, 2019). Safety problems remain open issues, that could be tackled by calculation of the backward reachable set.

The specific case of overparametrized networks, where there are way fewer training points than neurons per layer, and its ability to generalize are analyzed using Lyapunov techniques (Satpathi and Srikant, 2021). By studying the dynamics of the gradient descent, proofs of convergence are given. Proofs of stability and near-optimality for other learning algorithms that are value iteration can also be drawn using control tools (Granzotto et al., 2021). Recurrent neural networks stability is analyzed using advanced control tools (Ebihara et al., 2020). Taking a step backward, control tools such as observability and controllability also allow deriving impossibility results in AI (Brcic and Yampolskiy, 2021), i.e., put limits on what is possible to achieve with learning.

6. DISCUSSION ON RESEARCH PERSPECTIVES

In this section, we put in comparison the existing works and their stated motivation to the analysis presented in Section 3.3. Overall, most of the state of the art corresponds to expected promising directions of research combining ML and CT, with some being intensively explored, and others with some room for new results. There are however perspectives of combinations that we did not find in the state of the art, offering promising open research directions. Finally, some works realize patterns of combinations that were not motivated a priori by expected outcomes for both fields, but still lead to interesting state-of-the-art results.

6.1 Established combinations

Machine learning tools have been largely used to improve control identification, adaptation and tuning, with significant results thanks to their high accuracy RD1,2,6. Reversely, model-based approaches inspired from control theory have been successfully used in learning framework, thus significantly reducing exploration and training time RD9,14. Safety and robustness guarantees for ML are hot topics of research RD10,12, where maturity of the control tools are used on learning algorithms, see (Seidman et al., 2020) for an example on robustness properties. Finally, the use of control for training data selection is widely used, mostly in the robotics field RD16.

Despite many combination works existing aiming at the above-mentioned objectives, solid universal results are still missing, as well as democratization of those ideas to broader applications than robotics.

6.2 Emerging combinations

Recent works aim at using neural tools to enable CT to tackle systems with complex dynamics RD3, e.g., non-linear ones. Control based on neural network models is a promising research direction, that can moreover help scale control solution to larger dimension systems RD4.

While ML has been used to perform control checking, the systematic evaluation methodology using extensive real data has not yet been democratized among control theorists RD7. The ML statistical approach can be useful for evaluation of robustness or adaptation properties, in the cases where the system’s properties are unknown. The extension of control tools to broader objectives than stability of reference tracking is also a promising research direction, as have shown preliminary works RD5. The use of controllers to regulate ML’s hyperparameter also enables to reduce the train phase RD8.

In addition to stability and safety considerations, deterministic guarantees of ML algorithms accuracy is absolutely needed RD11. In a facial recognition example, mis-categorizing one face does not constitute a critical decision from the cyber-physical safety perspective as could be for instance a wrong braking command in an autonomous car, but could still be service-critical, e.g., in case of bank portal authorization systems. This motivation is related to the one regarding theoretical algorithm evaluation RD13,

classical consideration from the control community that could help improve learning theory.

6.3 Open directions

Despite all the works done at the intersection of control and learning, some benefits highlighted by Section 3.3 are—to the authors’ knowledge—so far fairly unexplored. ML algorithm adaptation has only started to be considered: given a time-variant system or a system with uncertainties or disturbances, how can an ML algorithm realize that its predictions are losing accuracy, and how to correctly adapt them? Work towards extending current ML techniques to continuously changing systems is needed, and could benefit from CT experience RD15.

A special attention can also be drawn to the initial moments of ML usage, particularly in the reinforcement learning scenario. As models and policies have to be constructed in real-time, online, initial decisions are very difficult to decide upon while they have a potential strong impact RD10,14. In mechatronics command for moving an object, for instance, choosing to move a robot arm forward or close its hand may either make the object fall (thus ruining the task) or successfully grab it. Without a priori knowledge, for instance using models, the failure probability is high.

Finally, ML’s explainability and decision traceability is a very hot topic, specially regarding citizen’s protection and fairness. Controllers are tools that enable to ensure those properties even when dealing with stochastic systems, combining ML and CT could enable to transfer explainability to ML RD17.

There are only few, to the best of our knowledge, interactions between ML and discrete control, possibly because of their different nature. Particularly, most research in Discrete Controller Synthesis essentially targets an exact formal computation; due to the nature of logic objectives, correctness is not amenable to approximations based on a notion of distance. A potential complementarity between RL and discrete control, addressing RD10,1,2, can take place at execution time of the discrete controllers. It can take the shape of the composition of a RL algorithm making proposals for configurations of a system, targeting optimally in performance metrics, while a discrete controller, guaranteeing essential safety properties, filters them in order to actuate only those that are correct, and inhibits the others in order to avoid inappropriate behaviors. Some relations can be drawn, however, at different levels.

6.4 Unexpected outcomes

In addition to the expected combination with regard to the motivations, extra categories of works can be found in the state of the art. ML has been used as a technique to adapt online a CT model, then used to tune the controller with a predefined law, see Section 4.1. While this approach does not a priori allow preserving guarantees of the resulting controller, it still could bring additional performance in many scenarios. Similarly, using ML algorithms as feedback controllers does not enable to guarantee their results (in terms of stability or robustness for instance) but could

still show good results in practice, as presented in Section 4.3. In non-critical use-cases in which guarantees are not necessary, those approaches may bring significant outcomes in terms of performance. Symmetrically, controllers have been used as an ML model optimization algorithm, see Section 5.3. While no guarantees can be transferred in this case, benefits of this composition can still be found in performance of the main task.

7. CONCLUSION

This work focuses on Neural-based Machine Learning and deterministic Control Theory. Despite being two substantially distinct fields, they have complementary challenges and strengths. The paper reviews their combinations, starting with existing works and deriving open directions. Firstly, advantages and drawbacks of each theory individually are presented, and their complementarity is analyzed. Learning shows very good performance with an ability to adapt to many use-cases, even of high complexity and scale, unlike control, however with limited safety. Control provides guarantees of stability, accuracy and robustness, from the very first instants, unlike learning, nonetheless with potentially suboptimal performances. Based on this analysis, research directions are given motivating neural-based learning and control combinations, such as guaranteeing learning performances and safety or enhancing control of non-linear systems. State of the art is then reviewed, with a proposed categorization regarding the combinations patterns. Distinction is done between works in which a learning component is used in a control framework (e.g., for modeling or for controller design) and those where control tools are integrated into a learning setup (e.g., to regulate training data or hyperparameters). The paper then takes a step backwards to discuss on the most-common combinations and their motivation, gives hints of research directions that would benefit from more efforts, and concludes on the unexpected benefits of control and learning combinations.

REFERENCES

- Abiodun, O.I. et al. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*.
- Abu-Khalaf, M. et al. (2021). Feedback from pixels: Output regulation via learning-based scene view synthesis. In *Learning for Dynamics and Control*.
- Aggarwal, C.C. et al. (2018). Neural networks and deep learning. *Springer*, 10, 978–3.
- Åkesson, B.M. and Toivonen, H.T. (2006). A neural network model predictive controller. *Journal of Process Control*, 16(9), 937–946.
- Alamir, M. (2020). On the use of supervised clustering in stochastic nmpe design. *IEEE TAC*.
- Alanwar, A. et al. (2021). Data-driven reachability analysis using matrix zonotopes. In *Learning for Dynamics and Control*.
- Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
- Amato, F. (2006). *Robust control of linear systems subject to uncertain time-varying parameters*. Springer.
- Balakrishnan, S. et al. (2008). Issues on stability of adp feedback controllers for dynamical systems. *IEEE Transactions on Systems, Man, and Cybernetics*.
- Balasubramaniam, G. (2019). Towards comprehensible representation of controllers using machine learning. In *IEEE/ACM International Conference on Automated Software Engineering*.
- Banerjee, S. and Chatterjee, A. (2019). Alera: Accelerated reinforcement learning driven adaptation to electromechanical degradation in nonlinear control systems using encoded state space error signatures. *ACM Transactions on Intelligent Systems and Technology*.
- Barreau, M., Aguiar, M., Liu, J., and Johansson, K.H. (2021). Physics-informed learning for identification and state reconstruction of traffic density.
- Basile, F. et al. (2020). An optimization-based approach to discover the unobservable behavior of a discrete-event system through interpreted petri nets. *IEEE Transactions on Automation Science and Engineering*.
- Behler, J. (2015). Constructing high-dimensional neural network potentials: A tutorial review. *International Journal of Quantum Chemistry*, 115(16), 1032–1050.
- Benosman, M. et al. (2019). Editorial for the special issue on learning-based adaptive control: Theory and applications. *International Journal of Adaptive Control and Signal Processing*.
- Bensoussan, A. et al. (2020). Machine learning and control theory. In *arXiv preprint*.
- Besançon, G. (2007). *Nonlinear observers and applications*, volume 363. Springer.
- Brcic, M. and Yampolskiy, R.V. (2021). Impossibility results in ai: A survey. *arXiv preprint arXiv:2109.00484*.
- Brosilow, C. and Joseph, B. (2002). *Techniques of model-based control*. Prentice Hall Professional.
- Buşoniu, L. et al. (2018). Reinforcement learning for control: Performance, stability, and deep approximators. *Annual Reviews in Control*.
- Caldas, R.D. et al. (2020). A hybrid approach combining control theory and ai for engineering self-adaptive systems. In *IEEE/ACM SEAMS*.
- Chen, R.T. et al. (2018a). Neural ordinary differential equations. *arXiv preprint*.
- Chen, S. et al. (2018b). Approximating explicit model predictive control using constrained neural networks. In *IEEE ACC*.
- Chen, X. and Hazan, E. (2021). Black-box control for linear dynamical systems. *Conf. on Learning Theory*.
- Chiuso, A. and Pillonetto, G. (2019). System identification: A machine learning perspective. *Annual Review of Control, Robotics, and Autonomous Systems*.
- Chu, S.Y. and Teng, C.C. (1999). Tuning of pid controllers based on gain and phase margin specifications using fuzzy neural network. *Fuzzy sets and systems*.
- Cong, S. and Liang, Y. (2009). Pid-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. *IEEE Transactions on Industrial Electronics*.
- Dadone, P. et al. (1998). Modeling and control of discrete event dynamic systems: a simulator-based reinforcement-learning paradigm. *International Journal of Intelligent Control and Systems*.
- Dean, S. and Recht, B. (2021). Certainty equivalent perception-based control. *Learning for Dynamics and Control*.
- Dean, S. et al. (2020a). On the sample complexity of the linear quadratic regulator. *Foundations of Computa-*

- tional Mathematics*.
- Dean, S. et al. (2020b). Robust guarantees for perception-based control. In *Learning for Dynamics and Control*.
- Dierks, T. and Jagannathan, S. (2009). Neural network output feedback control of robot formations. *IEEE Trans. on Systems, Man, and Cybernetics*.
- Donti, P.L. et al. (2020). Enforcing robust control guarantees within neural network policies. *CoRR*.
- Doyle, J.C. (1978). Guaranteed margins for lqg regulators. *IEEE TAC*.
- Doyle, J.C. et al. (2013). *Feedback control theory*. Courier Corporation.
- Dracopoulos, D.C. (1997). Genetic algorithms and genetic programming for control. In *Evolutionary algorithms in engineering applications*. Springer.
- Du, J. et al. (2020). Model-based reinforcement learning for semi-markov decision processes with neural odes. *arXiv preprint*.
- Duriez, T. et al. (2017). *Machine learning control-taming nonlinear dynamics and turbulence*. Springer.
- Ebihara, Y. et al. (2020). l_2 induced norm analysis of discrete-time lti systems for nonnegative input signals and its application to stability analysis of recurrent neural networks. *arXiv preprint*.
- Esteve, C. et al. (2020). Large-time asymptotics in deep learning. *arXiv preprint*.
- Fattahi, S. et al. (2019). Learning sparse dynamical systems from a single sample trajectory. In *IEEE CDC*.
- Fazel, M. et al. (2018). Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*.
- Fiechter, C.N. (1997). Pac adaptive control of linear systems. *Conf. on Computational learning theory*.
- Fliess, M. and Join, C. (2020). Machine learning and control engineering: The model-free case. In *Future Technologies Conference*. Springer.
- Franklin, G.F. et al. (2002). *Feedback control of dynamic systems*. Prentice hall.
- Fröhlich, L.P. et al. (2021). Cautious bayesian optimization for efficient and scalable policy search. In *Learning for Dynamics and Control*.
- Gelenbe, E. (1993). Learning in the recurrent random neural network. *Neural computation*, 5(1), 154–164.
- Ghai, U. et al. (2021). Generating adversarial disturbances for controller verification. In *Learning for Dynamics and Control*.
- Gilra, A. and Gerstner, W. (2018). Non-linear motor control by local learning in spiking neural networks. In *International Conference on Machine Learning*.
- Gradu, P. et al. (2020). Adaptive regret for control of time-varying dynamics. *arXiv preprint*.
- Granzotto, M. et al. (2021). When to stop value iteration: stability and near-optimality versus computation. In *Learning for Dynamics and Control*.
- Gros, S. and Zanon, M. (2019). Data-driven economic mpc using reinforcement learning. *IEEE TAC*.
- Gros, S. and Zanon, M. (2020). Reinforcement learning for mixed-integer problems based on mpc. *arXiv preprint*.
- Gros, S. et al. (2020). Safe reinforcement learning via projection on a safe set: How to achieve optimality? *arXiv preprint*.
- Guez, A. et al. (1988). Neural network architecture for control. *IEEE control systems Magazine*.
- Hazan, E. et al. (2020). The nonstochastic control problem. In *Algorithmic Learning Theory*.
- Hellerstein, J. et al. (2004). *Feedback control of computing systems*. John Wiley & Sons.
- Hespanha, J.P. (2018). Linear systems theory. In *Linear Systems Theory*. Princeton university press.
- Hewing, L. et al. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*.
- Hinrichsen, D. and Pritchard, A.J. (2005). *Mathematical systems theory I: modelling, state space analysis, stability and robustness*, volume 134. Springer.
- Hoffmann, H. (2015). Jouleguard: energy guarantees for approximate applications. In *Proceedings of the 25th Symposium on Operating Systems Principles*, 198–214.
- Hou, Z.S. and Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235, 3–35.
- Huang, C. et al. (2019). ReachNN: Reachability analysis of neural-network controlled systems. *ACM Trans. Embed. Comput. Syst.*
- Ivanov, R. et al. (2019). Verisig: verifying safety properties of hybrid systems with neural network controllers. In *ACM International Conference on Hybrid Systems: Computation and Control*.
- Jamshidi, P. et al. (2016). Fuzzy self-learning controllers for elasticity management in dynamic cloud architectures. In *ACM SIGSOFT Conference on Quality of Software Architectures*.
- Janny, S. et al. (2021). Deep kkl: Data-driven output prediction for non-linear systems. *arXiv preprint*.
- Jones, E.S. and Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Journal of Robotics Research*.
- Kamalapurkar, R. et al. (2016). Model-based reinforcement learning for approximate optimal regulation. In *Control of Complex Systems*. Elsevier.
- Kamthe, S. and Deisenroth, M. (2018). Data-efficient reinforcement learning with probabilistic model predictive control. In *International Conference on Artificial Intelligence and Statistics*, 1701–1710.
- Khalil, H.K. and Grizzle, J.W. (2002). *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ.
- Khan, S.G. et al. (2012). Reinforcement learning and optimal adaptive control: An overview and implementation examples. *Annual Reviews in Control*.
- Kirk, D.E. (2004). *Optimal control theory: an introduction*. Courier Corporation.
- Kirschner, J. and Krause, A. (2019). Stochastic bandits with context distributions. In *Advances in Neural Information Processing Systems*, 14113–14122.
- Kiumarsi, B. et al. (2017). Optimal and autonomous control using reinforcement learning: A survey. *IEEE Trans. on neural networks and learning systems*.
- Kolarijani, A.S. et al. (2018). Fast gradient-based methods with exponential rate: A hybrid control framework. In *International Conference on Machine Learning*.
- Korda, M. (2021). Stability and performance verification of dynamical systems controlled by neural networks: algorithms and complexity. *arXiv preprint*.
- Kretchmar, R.M. (2000). *A synthesis of reinforcement learning and robust control theory*. Colorado State University.

- Lagaris, I.E. et al. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*.
- Lama, P. and Zhou, X. (2013). Autonomic provisioning with self-adaptive neural fuzzy control for percentile-based delay guarantee. *ACM TAAS*.
- Landau, I.D. et al. (2011). *Adaptive control: algorithms, analysis and applications*. Springer.
- Larochelle, H. et al. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. *International Conf. on Machine learning*.
- Lauer, F. (2019). Error bounds for piecewise smooth and switching regression. *IEEE transactions on neural networks and learning systems*.
- Lazar, M. and Pastravanu, O. (2002). A neural predictive controller for non-linear systems. *Mathematics and Computers in Simulation*.
- Lee, C.H., Lee, Y.H., and Teng, C.C. (2002). A novel robust pid controllers design by fuzzy neural network. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, volume 2, 1561–1566. IEEE.
- Lee, C.C. (1990). Fuzzy logic in control systems: fuzzy logic controller. i. *IEEE Transactions on systems, man, and cybernetics*.
- Legat, B. et al. (2021). Abstraction-based branch and bound approach to q-learning for hybrid optimal control. In *Learning for Dynamics and Control*.
- Levine, S. and Koltun, V. (2013). Guided policy search. In *International Conference on Machine Learning*, 1–9.
- Li, Q. and Hao, S. (2018). An optimal control approach to deep learning and applications to discrete-weight neural networks. In *International Conf. on Machine Learning*.
- Li, Q. et al. (2017). Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*.
- Liu, L. et al. (2019). A review of industrial mimo decoupling control. *International Journal of Control, Automation and Systems*.
- Liu, Y. et al. (2021). A survey on evolutionary neural architecture search. *IEEE transactions on neural networks and learning systems*.
- Ljung, L. and Glad, T. (1994). *Modeling of dynamic systems*. Prentice-Hall, Inc.
- Lowrey, K. et al. (2018). Plan online, learn offline: Efficient learning and exploration via model-based control. *arXiv*.
- Lynen, S. et al. (2013). A robust and modular multi-sensor fusion approach applied to mav navigation. In *IEEE/RSJ conference on intelligent robots and systems*.
- Makdesi, A. et al. (2021). Data-driven abstraction of monotone systems. In *Learning for Dyn. and Control*.
- Martin, C.H. et al. (2021). Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*.
- Matni, N. et al. (2019). From self-tuning regulators to reinforcement learning and back again. In *IEEE CDC*.
- Mayne, D. (2000). Nonlinear model predictive control: Challenges and opportunities. *Nonlinear model predictive control*, 23–44.
- Mhammedi, Z. et al. (2020). Learning the linear quadratic regulator from nonlinear observations. *arXiv preprint*.
- Miller, W.T. et al. (eds.) (1990). *Neural networks for control*. MIT Press.
- Mishra, N. et al. (2018). Caloree: Learning control for predictable latency and low energy. *ACM SIGPLAN Notices*.
- Moe, S. et al. (2018). Machine learning in control systems: An overview of the state of the art. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer.
- Moghadam, R. and Lewis, F.L. (2019). Output-feedback h_∞ quadratic tracking control of linear systems using reinforcement learning. *International Journal of Adaptive Control and Signal Processing*, 33(2), 300–314.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.
- Mordatch, I. and Todorov, E. (2014). Combining the benefits of function approximation and trajectory optimization. *Robotics: Science and Systems*, 4.
- Morimoto, J. and Doya, K. (2005). Robust reinforcement learning. *Neural computation*, 17(2), 335–359.
- Nagabandi, A. et al. (2018). Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE ICRA*.
- Neftci, E.O. and Averbeck, B.B. (2019). Reinforcement learning in artificial and biological systems. *Nature Machine Intelligence*, 1(3), 133–143.
- Newton, M. and Papachristodoulou, A. (2021). Exploiting sparsity for neural network verification. In *Learning for Dynamics and Control*, 715–727. PMLR.
- Novak, R. et al. (2018). Sensitivity and generalization in neural networks: an empirical study. *arXiv*.
- Parisi, G.I. et al. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*.
- Pauli, P. et al. (2021). Offset-free setpoint tracking using neural network controllers. In *Learning for Dynamics and Control*.
- Pioveso, M.J. and Owens, A.J. (1991). Neural network process control. In *Proceedings of the conference on Analysis of neural network applications*, 84–94.
- Poveda, J.I. et al. (2019). Hybrid online learning control in networked multiagent systems: A survey. *International Journal of Adaptive Control and Signal Processing*.
- Radac, M.B. and Precup, R.E. (2016). Three-level hierarchical model-free learning approach to trajectory tracking control. *Engineering Applications of Artificial Intelligence*, 55, 103–118.
- Rajamäki, J. and Hämäläinen, P. (2017). Augmenting sampling based controllers with machine learning. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.
- Rantzer, A. (2018). Concentration bounds for single parameter adaptive control. In *IEEE ACC*.
- Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*.
- Ren, P.o. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys*.
- Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.
- Ruiz-Balet, D. and Zuazua, E. (2021). Neural ode control for classification, approximation and transport. *arXiv*.
- Samy, I. et al. (2010). Neural-network-based flush air data sensing system demonstrated on a mini air vehicle. *Journal of aircraft*.

- Sarangapani, J. (2006). *Neural network control of nonlinear discrete-time systems*. CRC press.
- Sarkar, T. and Rakhlin, A. (2018). How fast can linear dynamical systems be learned? *arXiv preprint*.
- Sarkar, T. et al. (2019). Finite-time system identification for partially observed lti systems of unknown order. *arXiv preprint*.
- Satpathi, S. and Srikant, R. (2021). The dynamics of gradient descent for overparametrized neural networks. In *Learning for Dynamics and Control*, 373–384. PMLR.
- Sawant, S. and Gros, S. (2022). Bridging the gap between qp-based and mpc-based rl. *arXiv preprint*.
- Schaul, T. et al. (2013). No more pesky learning rates. In *International Conference on Machine Learning*.
- Seidman, J.H. et al. (2020). Robust deep learning as optimal control: Insights and convergence guarantees. *arXiv preprint*.
- Sidrane, C. and Kochenderfer, M.J. (2019). OVERT: Verification of nonlinear dynamical systems with neural network controllers via overapproximation. *Safe Machine Learning workshop at ICLR*.
- Silver, D. (2015). UCL Course on RL. URL <https://www.davidsilver.uk/teaching/>.
- Simchowitz, M. et al. (2018). Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*.
- Sirignano, J. and Spiliopoulos, K. (2018). Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375, 1339–1364.
- Stengel, R.F. (1991). Intelligent failure-tolerant control. *IEEE Control Systems Magazine*, 11(4), 14–23.
- Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Sze, V. et al. (2017). Efficient processing of deep neural networks: A tutorial and survey. *CoRR*.
- Szegedy, C. et al. (2013). Intriguing properties of neural networks. *arXiv preprint*.
- Tadiparthi, V.V. and Bhattacharya, R. (2022a). Data-driven verification using efficient active learning. In *AIAA SCITECH 2022 Forum*, 2086.
- Tadiparthi, V.V. and Bhattacharya, R. (2022b). Estimating invariant sets using physics-informed neural networks. In *AIAA SCITECH 2022 Forum*, 1441.
- Tang, S. et al. (2018). Aggressive flight with suspended payloads using vision-based control. *IEEE Robotics and Automation Letters*.
- Thorpe, A.J. et al. (2021). Learning approximate forward reachable sets using separating kernels. In *Learning for Dynamics and Control*.
- Tokic, M. (2010). Adaptive ϵ -greedy exploration in reinforcement learning based on value differences. In *Annual Conference on Artificial Intelligence*, 203–210. Springer.
- Tran, H.D. et al. (2020). NNV: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In *Computer Aided Verification*. Springer.
- Tu, S. and Recht, B. (2019). The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint. In *Conference on Learning Theory*, 3036–3083. PMLR.
- Ushio, T. and Yamasaki, T. (2003). Supervisory control of partially observed discrete event systems based on a reinforcement learning. In *IEEE International Conference on Systems, Man and Cybernetics*.
- Van De Ven, P.W. et al. (2007). Neural network augmented identification of underwater vehicle models. *Control Engineering Practice*.
- Van Otterlo, M. and Wiering, M. (2012). Reinforcement learning and markov decision processes. In *Reinforcement Learning*, 3–42. Springer.
- Vrabie, D. et al. (2009). Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*.
- Weinan, E. (2017). A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1), 1–11.
- Weyns, D. et al. (2021). Towards better adaptive systems by combining mape, control theory, and machine learning. In *IEEE SEAMS*.
- Williams, G. et al. (2017). Information theoretic mpc for model-based reinforcement learning. In *ICRA*. IEEE.
- Wonham, W.M. (2015). Supervisory control of discrete-event systems. *Springer*.
- Wu, Z. et al. (2019). Machine learning-based predictive control of nonlinear processes. part i. *AIChE Journal*.
- Xu, F. et al. (2019). Explainable ai: A brief survey on history, research areas, approaches and challenges. In *CCF international conference on natural language processing and Chinese computing*. Springer.
- Yaghoubi, S. and Fainekos, G. (2019). Gray-box adversarial testing for control systems with machine learning components. In *CM International Conference on Hybrid Systems: Computation and Control*.
- Yamasaki, T. and Ushio, T. (2004). Decentralized supervisory control of discrete event systems based on reinforcement learning. *IFAC Symposium on Large Scale Systems 2004: Theory and Applications*.
- Yan, H. et al. (2019). On robustness of neural ordinary differential equations. *arXiv preprint*.
- Zamarreño, J. and Vega, P. (1999). Neural predictive control. application to a highly non-linear system. *Engineering Applications of Artificial Intelligence*.
- Zanon, M. and Gros, S. (2019). Safe reinforcement learning using robust mpc. *arXiv preprint arXiv:1906.04005*.
- Zhang, H. et al. (2018). A learning-based synthesis approach to the supremal nonblocking supervisor of discrete-event systems. *IEEE TAC*.
- Zhang, N. and Capel, N. (2021). Leoc: A principled method in integrating reinforcement learning and classical control theory. In *Learning for Dyn. and Control*.
- Zhao, Z. et al. (2020). Event-based control for online training of neural networks. *IEEE Control Sys. Letters*.