IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 4, 2022, accepted March 14, 2022, date of publication March 21, 2022, date of current version March 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3160709

Toward Data-Driven Optimal Control: A Systematic Review of the Landscape

KRUPA PRAG[1], MATTHEW WOOLWAY[2], AND TURGAY CELIK[3,4]

[1]School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg 2000, South Africa
[2]Faculty of Engineering and the Built Environment, University of Johannesburg, Johannesburg 2006, South Africa
[3]School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg 2000, South Africa
[4]Wits Institute of Data Science, University of the Witwatersrand, Johannesburg 2000, South Africa

Corresponding authors: Krupa Prag (krupa.prag@gmail.com) and Turgay Celik (celikturgay@gmail.com)

ABSTRACT This literature review extends and contributes to research on the development of data-driven optimal control. Previous reviews have documented the development of model-based and data-driven control in isolation and have not critically reviewed reinforcement learning approaches for adaptive data-driven optimal control frameworks. The presented review discusses the development of model-based to model-free adaptive controllers, highlighting the use of data in control frameworks. In data-driven control frameworks, reinforcement learning methods may be used to derive the optimal policy for dynamical systems. Attractive characteristics of these methods include not requiring a mathematical model of complex systems, their inherent adaptive control capabilities, being an unsupervised learning technique and their decision-making abilities, which are both an advantage and motivation behind this approach. This review considers previous reviews on these topics, including recent work on data-driven control methods. In addition, this review shows the use of data to derive system dynamics, determine the control policy using feedback information, and tune fixed controllers. Furthermore, the review summarises various data-driven methods and their corresponding characteristics. Finally, the review provides a taxonomy, a timeline and a concise narrative of the development of model-based to model-free data-driven adaptive control and underlines the limitations of these techniques due to the lack of theoretical analysis. Areas of further work include theoretical analysis on stability and robustness for data-driven control systems, explainability of black-box policy learning techniques and an evaluation of the impact of the extension of system simulators to include digital twins.

INDEX TERMS Data-driven control, adaptive control, model-free, model-based, model predictive control, optimal control, learning-based control, systematic review.

I. INTRODUCTION

Control systems regulate the behaviour of various industrial systems by providing a control response to the system's current state. These regularisations referred to as the control policy within control frameworks, state the prohibitions and permissions of the system and govern the actions taken by the controller. Simply, the control policy maps states to actions. Note that the system is also referred to as the plant, and is interchangeably used with the term system throughout this paper.

The field of control theory is vast, with fast-evolving literature and process controller designs. Traditionally, the design of controllers employed by various industrial systems has been model-based. Hence, the control policy is dependent on the mathematical model representing the physical system's dynamics to determine the actuation to be applied to the system, given its current state.

Before developing and studying model-free adaptive control frameworks, model-based frameworks were extended to complex and nonlinear systems by making assumptions to simplify the task of encapsulating and modelling both the system's physical dynamics and the experienced external disturbances. However, these approximations and simplifications are not practical and restrict the performance of these systems. Due to the challenges which come with precisely modelling complex systems, model-dependent control systems are neither widely applicable nor feasible [1]. Furthermore, fixed controllers, used in primitive control systems, employ a predefined control policy that is applied to the system irrespective of any changes experienced. It is highlighted that control frameworks with fixed controllers and

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Alshabi.

model-dependent control systems are not efficient in achieving the goal of adaptive control, which requires handling uncertainties or disturbances and predicting scenarios beyond the classified objects in the operational environment whilst prioritising safety [2], [3]. More recently, Data-Driven Control (DDC) frameworks have been developed to address some of these feats and have become prominent, given both the explosion of available data in various industries and the accessibility to the computational power of modern-day computers.

While the limitations and drawbacks of fixed controllers and model-based control frameworks led to the development of DDC frameworks, there is no mutual exclusivity in utilised methods. DDC methods may be model-independent or used to enhance model-dependent control systems. For example, some DDC frameworks which directly use only the input-output (I/O) information of a system to determine a control policy using learning-based or iterative techniques are independent of the considered system's mathematical model. This model-independence is both an advantage, and a motivation behind model-free control frameworks [4]. The many uses and objectives for using data for control systems include: developing a controller for a model-free framework, system identification, construction of stochastic or uncertainty models, and tuning of fixed controllers [1]. It is important to highlight that model-free control is a particular application of DDC methods, as a data-driven technique is used to extract the need for a mathematical model representation of a system by either deriving the policy directly from the I/O data of the system, or for identifying the system. Whilst the construction of stochastic disturbance model and tuning of fixed controllers use data-driven techniques in conjunction with model-based control frameworks.

Furthermore, another reason for the development of DDC frameworks is to construct an adaptive optimal control policy that finds a control strategy for a dynamical system over some time such that the objective function is optimised and the control policy evolves to adapt to changes. Reinforcement Learning (RL) is an example of an iterative unsupervised learning-based method with the inherent characteristics of adaptability, which is contrasting and advantageous compared to past fixed-controllers. These capabilities highlight the evolution and advancements of the process of controller designs. However, drawbacks include that the stability analysis of these methods is primitive and a formidable challenge [5], [6], and that during the exploration phase of determining the control policy, the RL agent may apply actions that do not satisfy the action constraints which may leave the safety of these techniques questioned. These learning-based techniques require modern-day compute power to provide realistic and computationally efficient responses to online feedback signals. Another promising research avenue is the synergy of model-based frameworks and data-driven learning-based techniques, which is further discussed by [7] and in this review. An overview of the use of data in control systems is given in Fig. 1.

This literature review summarises the use of data in control systems. The primary objective is to provide a concise narrative of the development timeline and taxonomy from traditional model-based control systems to model-free data-driven optimal adaptive control frameworks. This study hopes to provide a single review of DDC techniques which can be used by both intelligent control and RL communities.

The main challenge highlighted in previous autonomous control reviews to date is to develop a control framework that is robust to disturbances, such that the system converges to the desired target within a minimum time and for stability to be maintained. These are pertinent to address in the safety of the operation of these systems [3]. This literature review points to literature on related work to analyse control techniques and emerging directions in this field.

The literature review is structured as follows. Section II gives a brief technical introduction of the classification and terminology of control frameworks. Section III details the methodology and the procedure used to conduct this literature survey. Section IV describes the timeline and taxonomy of control systems from their primitive stages to current novel data-driven control techniques. A description and the development of model-based and model-free control frameworks are respectively discussed in Section V and Section VII, while controller tuning techniques are discussed in Section VI. Section VIII discusses emerging trends in this area of research, while Section IX draws some final conclusions.

II. TERMINOLOGY AND CLASSIFICATION OF CONTROL SYSTEMS

This section considers key concepts and terminology of control theory and highlights essential features on which control systems are classified. These characteristics include the number of inputs and outputs of a system, the type of I/O data, the techniques used by the controller, and the configuration of the information used by the controller.

A. CONTINUOUS-TIME AND DISCRETE-TIME CONTROL SYSTEMS

Based on whether the signal used in a control system is continuous or discrete determines whether the control system is a continuous-time or discrete-time system. A continuous-time control system has all the system's variables defined as a function of time. Conversely, if the system variables are defined at distinct discrete-time steps, then the system is a discrete-time control system [8].

B. SISO AND MIMO CONTROL SYSTEMS

Single Input and Single Output (SISO) control systems have one input and have one output signal. Whereas systems that have more than one input and more than one output are called Multiple Input and Multiple Output (MIMO) control systems [9].

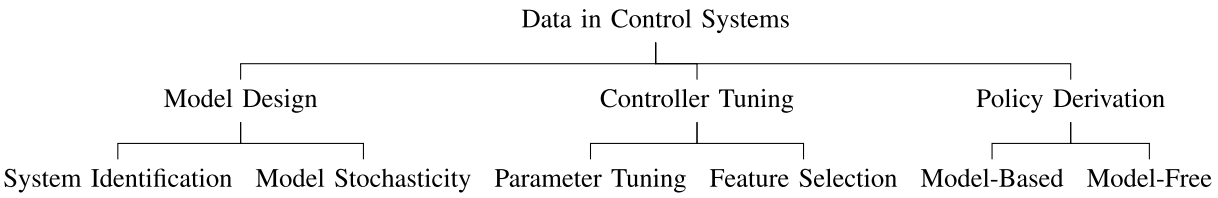


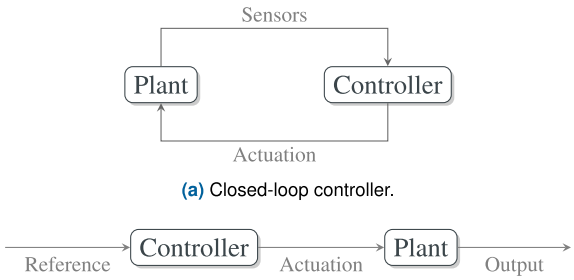
FIGURE 1. Overview of the use of data in control frameworks.

C. OPEN-LOOP AND CLOSED-LOOP SYSTEMS

Based on the feedback path, a control system is classified as either an open-loop or closed-loop control system [9]. If the output or feedback from the system is not used in determining the next control action, the system is classified as an open-loop control system. However, if the output or feedback from the system is fed-back to the actuator to be used in determining the actuation to be applied, then the system is said to be a closed-loop control system.

In contrast to closed-loop controllers, open-loop controllers are considered easier to construct, however, they are unreliable given that they do not have a feedback mechanism and are thus unable to remove the impact of disturbances using the feedback information. Furthermore, while closed-loop controller feedback mechanisms are advantageous for providing better accuracy and reducing the impact of noise, these control systems' construction is relatively complex.

A high level overview of a feedback control and an open-loop system are given respectively in Fig. 2a and Fig. 2b. Where the controller is designed using learning control techniques, without any details on the considered plant, and merely based on the sensors' readings, the actuator or controller should determine the actuation to be applied to the plant.



(a) Closed-loop controller.

(b) Open-loop controller.

FIGURE 2. A high level overview of closed-loop and open-loop control systems.

D. LINEAR AND NONLINEAR CONTROL SYSTEMS

A linear control system obeys the superposition theorem [10], the system is governed by linear differential equations, and the output or the response varies linearly with respect to the input or the actuation. In contrast, nonlinear systems do not necessarily satisfy the superposition principle, the system is governed by equations of nonlinear nature, and the outputs do not vary in a linear relationship with respect to the input [11], [12].

E. MODEL-BASED AND MODEL-FREE CONTROL FRAMEWORKS

Model-based control systems use the physical dynamics of the system's structure, given in the form of a mathematical representation, in determining the actuation signal to be applied to the system. In contrast, model-free control systems use linearisation techniques and learning-based techniques to develop a controller based on historical data or the output of the plant at each iteration and not on any assumptions on the system model [13].

F. ONLINE AND OFFLINE DATA

Offline measured data is a historical data set, whilst online measured data is the information obtained from real-time channels. Online measured data allow for real-time updates, whilst the usage of offline measured data requires regular updates to account for new trends that can be seen in recently obtained measurements.

G. FIXED CONTROL AND ADAPTIVE CONTROL

A fixed control system has a predefined control architecture that is used in determining actuations to apply to the plant irrespective of any changes in the environment. However, in contrast, an adaptive control system adjusts the control method with respect to the control system's parameters. There are two particular adaptive control categories: direct and indirect. The direct adaptive methods directly respond to the output of the plant, thereby iteratively updating the control policy or the mapping of the I/O data. Indirect adaptive control methods estimate the parameters of the plant and use the estimated model to adjust the controller by fine-tuning the controller's parameters [14], [15].

H. ON-POLICY AND OFF-POLICY METHODS FOR REINFORCEMENT LEARNING

RL algorithms may be considered as on-policy or off-policy methods. On-policy methods uniformly update and improve the implemented policy that is used to determine the controller's actuations to be applied to the system.

State-Action-Reward-Action (SARSA) is an example of an on-policy method where the behavioural and target policy are the same. Which means that the agent learns directly from its experiences. When the behavioural and target policy are

the same, the agent both selects the actuation and uses the selected actuation.

Off-policy methods have more freedom in exploring the environment than on-policy methods. Off-policy methods update the policy by merely estimating future rewards and actions given by the generated data and are independent of the agent's actions [16], [17]. In contrast to on-policy methods, in off-policy methods, the agent does not select its own actuations but instead learns from exploration. Hence the target policy and the behavioural policy are different. Q-learning is an example of an off-policy method.

III. METHODOLOGY

The structure of this systematic literature review is primarily based on the guidelines provided by [18]. The review studies the use of data in control systems with a particular focus on the development of data-driven methods for model-free adaptive control. The literature review gives an overview of the uses of data in control system frameworks, the timeline and taxonomy of the development of controller designs from model-dependent designs to model-independent designs, reviews common adaptive control techniques and underlines their strengths and weaknesses, discusses the limitations of the literature, and indicates recent advances and emerging directions.

To keep the narrative of the development on control techniques from primitive stages to current control frameworks concise, this study only includes common data-driven control techniques and omits hybrid techniques used. The reviewed methods are critically discussed to highlight both their applications and limitations. As summarised in the introduction in Fig. 1, the three categories of data in control systems are model design, controller tuning and policy derivation. These are common in similar literature surveys [19], [20]. Other classifications include adaptive or fixed controllers, but these topics have been discussed alongside model-based, and model-free controllers as this literature review focuses on adaptive controllers. Other characteristics on which control systems are classified include switching mechanisms and whether the control framework is a distributed system. The use of historical data in improving model design includes system identification and modelling of the stochasticity experienced by a plant. Methods considered for controller parameter tuning using data include Iterative Feedback Tuning (IFT), Virtual Reference Feedback Tuning (VRFT), Correlation Based Tuning (CBT) and non-Correlation Based Tuning (nCBT). Literature on the use of data in feature selection used in feedback control is pointed to. The primary focus of this survey is the use of data in policy derivation. The model-based techniques consider Model Predictive Control (MPC) and its data-driven extension Data-driven Model Predictive Control (DDMPC); Model-free Adaptive Control (MFAC) techniques include Iterative Learning Control (ILC), Lazy Learning (LL), Dynamic Linearisation Techniques (DLT) and prominent RL based methods such as Deep Q-network (DQN), Deterministic Policy Gradient (DPG), Deep DPG (DDPG), Trust Region Policy Optimisation (TRPO), and Proximal Policy Optimisation (PPO). This review points to other intelligent control techniques such as Bayesian probability, fuzzy logic and evolutionary computation used in control frameworks in Section IV, however, it focuses on iterative learning-based methods which may or may not be neural network (NN) based for model-free policy derivation techniques. Furthermore, the narrative is centred around discrete-time control systems, given their predominance as they are easier to integrate, have a lower computational cost than continuous-time control computations, and have a more comprehensive range of developed algorithms available to solve problems of this nature [21].

The literature appraisal and selection process entailed using the following search words: 'Data-driven control', 'Model-free adaptive control using data-driven techniques', 'Data-driven model predictive control', 'Intelligent control', and 'Learning-based control'. Publications between 2011-2021 from peer-reviewed journals and conference proceedings were considered. It must be noted that the Background, Section IV, includes earlier works dating back to the late 19^{th} century. Furthermore, textbooks considered were not restricted to the mentioned time frame.

Searches for literature on the aforementioned keywords were performed in Google Scholar, Web of Science, IEEE Xplore, Science Direct, Annual Reviews and Springer Link. From the returned results after searching the aforementioned keywords in the various databases, survey papers were first perused and then other returned articles' abstracts, introductions and conclusions were analysed. Articles that gave insight on topics considered under data in control were read in their entirety and included in this survey. Finally, only articles written in the English language were only considered.

A. RECENT LITERATURE SURVEYS AND DEVELOPMENTS ON DATA USED IN CONTROL SYSTEMS

Seminal literature surveys on data-driven methods for control methods were seen from the late 20^{th} century. Table 1 highlights the main contributions and topics discussed in the respective literature surveys conducted between 2011-2021 that are closely related to this review. These survey papers were seeds in searching for literature included in this review. The main contributions of this review are also included in Table 1. This literature review aims to provide a single review that can be referenced by both the robotics and automatic or intelligent control communities to discuss the various uses of data for optimal and adaptive control. Data in control for the various topics such as controller tuning, and both NN based and non-NN based frameworks have been reviewed [19]. Furthermore, this review provides an extension and contribution to developments since 2018 and provides a timeline and taxonomy of both control frameworks, which are dependent and independent of the model dynamics.

Data usage in both model-based and model-free frameworks are respectively discussed in Section V and VII, an overview of controller tuning using data-driven techniques

TABLE 1. Overview of the recent literature survey papers on the development of data-driven control.

Reference	Model-Based	Model-Free	Controller Tuning	Non-NN Based	NN-Based	Objectives
[20] (2013)	✓	✓	✓	✓		• Highlights the challenges and relationship of model-based control and applications of state of the art DDC methods. • An overview is given of model-based and model-free control techniques in designing the controller. The use of data for both controller design and controller tuning are summarised. • Distinguishes and classifies the various methods based on their characteristics.
[19] (2018)	✓	✓	✓	✓	✓	• Classifies adaptive control techniques based on whether or not the methods are model-based or data-driven and describes the approach used in deriving the policy. • Model-based approaches include a discussion on 'adaptive regulation' considers unknown disturbances without explicitly modelling the system. • Using learning-based techniques to improve controllers in a model-based framework.
[1] (2018)	✓			✓		• Proposed formulations are given of the use of data to formulate uncertainty in the model design. Prediction models, which include environment models that are used to navigate and manipulate objects, can be deterministic, stochastic or scenario-based. • The review includes an overview of predictive control frameworks. An analysis of these methods discuss ensuring recursive feasibility, convergence, robustness, constraint satisfaction, and computational tractability are discussed. • The impact of the prediction horizon length is analysed. • The properties of linear MPC state-feedback policies with or without disturbances are presented.
[22] (2019)		✓			✓	• Gives an overview of the recent progress of RL for process control. • Highlights best-suited systems and underlines constraints or limitations of the various applications. • Compares the characteristics of MPC to RL.
[23] (2020)		✓			✓	• Highlights, critically compares and reviews RL methods used in process control. • Classifies the various RL methods. • Underlines the shortcoming of RL, which include un-established stability theory and not accounting for constraints in model-free frameworks.
Our Work (2022)	✓	✓	✓	✓	✓	• Provides a detailed chronological description of the evolution of control frameworks from primitive model-based techniques to DDC techniques for adaptive optimal control. • Provides a single review of reference which is used to discuss the ensemble of data uses in control systems, with a primary focus on policy derivation from data. • Points to recent works on the development of theoretical analysis of model-free methods which include studies and formalisation on stability, robustness and convergence. Furthermore, the need for studies on the explainablity and interpretability of black-box algorithms. • Points to emerging directions of work in this field of data-driven optimal control including the requirement of development of high fidelity simulators to be used in the process of agent training and, the digital twin to optimise the end-to-end process of the development of control frameworks.

is given in Section VI, and emerging trends are presented in Section VIII.

IV. BACKGROUND

From 1760 to 1840, society's once agrarian-handicraft economy slowly transitioned to one dominated by mechanised factory systems and machine tools, hence transforming societies to be more industrialised and urban. In modern history, this transition period is known as the Industrial Revolution. During the late 18^{th} to 19^{th} century, the industrial sector had not only become fast-growing but had also initiated making adaptations of the available technology. This initial progress was shortly followed by analysing the designs of continuously operating process systems to improve and optimise their performance. Various attempts at maintaining accurate control of these dynamical systems led to both practical and theoretical development being done in the field of Control Theory, as first proposed by [24]. The reader is referred to the survey paper [25], [26] on the early progression of control theory.

Control systems or control engineering is a discipline that practically applies control theory to design systems with desired behaviours in a given environment. *Control systems* can be formally described as a device that generates autonomous behaviour through computation and actuation [1]. *Feedback systems* are a particular process that may form a part of control systems to improve the performance of control systems by returning the output of the system to be utilised as a part of the system's input. Feedback controllers were widely used in the early years of the 20^{th} century for voltage, current, and frequency regulation; boiler control for steam generation; electric motor speed control; ship and

aircraft steering and auto stabilisation; and temperature, pressure, and flow control in the process industries [25]. As a result, the controllers' design were tailored specifically to these applications. However, most of these controllers were designed without a thorough understanding of the control system's dynamics and the actuating control device. This lack of understanding was due to poor theoretical backing at the time, with no common language to discuss these types of problems. Fortunately, since the control systems applications were simple regulations, the undeveloped theoretical rigour was not detrimental. Although, there were more complex mechanisms involving complicated control laws which were being developed, such as the automatic ship-steering mechanism devised by Elmer Sperry in 1911, which incorporated Proportional Integral Derivative (PID) control and automatic gain adjustment to compensate for the disturbances caused when the sea conditions changed [25].

During World War I (1914-1918), major developments emerged in stochastic systems, including the fire control work done by [27]. From 1935 to 1940, advances in the understanding of control system analysis and design were being made by several independent groups around the globe. However, the beginning of the transition period leading to the formalisation of modern control theory took form after the conference on ''Automatic Control'' held in July 1951 at Cranfield, England, and the ''Frequency Response Symposium'' held in December 1953 in New York [28].

The wartime experience during World War II (1939-1945) demonstrated the power of the frequency response approach to the design of feedback systems and revealed the weaknesses of any design method based on the assumption of linear deterministic behaviour. The two assumptions which facilitate control algorithm design are: there is no human interaction with the system, and precise knowledge of the environment is known with which the system interacts [1]. However, these are not practical assumptions when considering industry scenarios. The nature of real systems are not necessarily linear, real measurements contain errors and are contaminated by noise, and in real systems, both the process and the environment are uncertain. In order to have the best-suited controller for the system, the design techniques to be used should consider the following behaviours: linear and nonlinear, deterministic and non-deterministic, and the presence of noise or measurement error. In the 1980s, post World War II, research had begun to make optimal feedback logic more robust to disturbances and variations in the measurements received from the systems [26], [29]. This research topic has rapidly grown since its conception and is still a topic of research to date.

The development of control system frameworks through key historical events such as the Industrial Revolution, World War I, and World War II highlighted the absence of systematic methods to handle hard constraints imposed on control systems. This had resorted to ad-hoc methods, such as single loop controllers augmented by various selectors. The birth of MPC had brought about a means to accommodate the requirement of having controllers take imposed constraints into account. MPC has a predictive capability and can better encapsulate dynamic characteristics of dynamical systems than traditional PID controllers. In addition, adaptive control techniques were developed to account for uncertainties or adaptations of control systems, with these being either model-based or data-driven approaches [19].

The evolution of controller objectives and designs are highlighted in this section. In summary, initially, control systems performed predefined actions based on the system's current state response. However, this objective was satisfied by fixed controllers as the understanding of control theory grew and developed model-based techniques to encapsulate the system dynamics better. Due to the complexities of the considered plants, the possibility of accounting for disturbances brought about the idea of model-free adaptive control. Essentially, none of these methods is explicitly model-free as the system dynamics are captured through various function approximation methods. These methods include traditional statistical methods and learning-based methods.

Intelligent probabilistic and statistical methods include fuzzy logic [30], [31], Kalman filters, particle filters [32], [33], Bayesian optimisation [34], amongst others. Since the conception of the fuzzy logic method, stability analysis of the technique has been formalised, it has been applied in both model dependent and independent control frameworks and has been applied to problems in a range of industries [35]. The reader is referred to the following surveys and applications of this technique to control problems [36]–[39]. Although fuzzy logic in control theory has shown success in several applications, unfortunately, in some cases, its drawback limits the application to control systems. Fuzzy logic drawbacks include it not being considered a systemic approach to solving problems, inconsistent performance, and significant training and validation requirements. There are multiple applications of Kalman filters to control problems, as reviewed in [40], as they are computationally efficient in terms of memory use. However, they assume that both the system and the observations are linear. Bayesian optimisation has been directly applied to control problems and used an optimisation technique for hyper-parameter tuning. Bayesian optimisation is sensitive to the parameters used, and the difficulty of estimating the Bayesian optimisation model is itself a drawback.

This literature review focuses on the development from MPC, DDMPC, and learning-based model-free adaptive control techniques. These are further discussed in this section and this paper.

A. MODEL PREDICTIVE CONTROL

MPC is a feedback control algorithm that uses the model representation to forecast behaviours by solving an online optimisation problem to select the most suitable control action, such that the system being acted upon (plant or process) is driven towards the desired target. This advanced model-based process control method was born in the petrochemical industry in the late 20^{th} century. This class of

model-based control methods require an explicit dynamic model of the plant to predict the impact of future actuations of the control variables based on the feedback or output from the plant. MPC is commonly known as Receding Horizon Control (RHC) as, in brief, at each discrete time step, the future actuations to be applied to the plant are determined. This set of actuations is obtained using the dynamic model, and at each sampling time, the set of future actuations is updated based on the updated feedback from the system. For details on the early development of MPC, the reader is referred to the survey paper [41]. MPC is a MIMO advanced process control method, whilst the PID controller is traditionally SISO, however, has been extended and applied to MIMO systems. Furthermore, the ability to acknowledge constraints and the predictive capability of the MPC framework are seen as an improvement and advantages in comparison to traditional PID controllers. In contrast, PID controllers are model-free in comparison to model-based MPC frameworks.

The statement made in [42] encapsulates the objectives of MPC: ''One technique for obtaining a feedback controller synthesis from knowledge of open-loop controllers is to measure the current control process state and then compute very rapidly for the open-loop control function. The first portion of this function is then used during a short time interval, after which a new measurement of the process state is made and a new open-loop control function is computed for this measurement. The procedure is then repeated.''. This statement guided the development of the family of MPC controller designs into mature techniques to tackle control problems in the industry with a strong theoretical basis. The MPC model was designed to solve multi-variable, constrained, infinite horizon, and possibly nonlinear optimal control problems via finite horizon solutions with a receding horizon implementation. These finite horizon solutions involve optimising the objective function for the (finite) prediction horizon, where the predictions are based on a mathematical model of the dynamical system to be controlled in real-time. Some of the most primitive work on MPC, which laid the foundation of this field, and the applications of MPC in industry include the description of successful applications of Model Predictive Heuristic Control (MPHC) in 1978 [43] which wss later known as Model Algorithmic Control (MAC), and the outline of Dynamic Matrix Control (DMC) [44], [45]. Both algorithms, MAC and DMC, make explicit use of dynamic process model.

Having a theoretical foundation set up for MPC in the late 20^{th} century, the early 2000*s* focus was on the development of the MPC controller design to reduce orders of magnitude of computation time to compute online optimisation efficiently. Such real-time responses could be given to the technology to which it was applied, thus requiring fast-sampling rates. Initially, explicit MPC control laws were determined offline to achieve speed up through a customised algorithm, which proved to be orders of magnitudes faster than the generic solver. However, as the horizon size or states and constraints increased, the number of polyhedral regions scaled, making the lookup task in a table difficult to implement in practice. Hence, [46] proposed methods include a combination of table storage and online optimisation, or simplifying the problem by imposing equality constraints as proposed in [47], or using approximate primal barrier interior point method adorned with several customised features like fast Newton step computation and a fixed barrier parameters as suggested in [48]. The online approach is imperative and provides an added advantage of weighted parameters horizon size on model parameters which can be changed as required, unlike explicit methods where entirely new lookup tables would have to be constructed.

Given the potential of MPC, it has been widely applied to applications including fields of power electronics [49], [50], data centre cooling [51]–[53] and unmanned autonomous vehicles (UAVs) [54] amongst others. The reader is referred to [55] for a detailed review on the development of MPC. In Section V-A a detailed description of the MPC method is given.

B. DATA-DRIVEN CONTROL

In recent years, information has been available in abundance. For example, data or information recorded from plants have been used to model system dynamics, design stochastic models representing noise [1], fine-tune controllers [20], and derive the control law merely using I/O data and learning methods [19], [20]. Control frameworks that use data-driven approaches may be applied to model-based or model-free systems and may use either or both online and offline data. The definition of DDC varies throughout the evolution of this field and in the literature. In some instances, DDC refers to a model-free framework that use data with intelligent algorithms to derive the control policy. In contrast, in other instances, DDC refers to the general use of data in control irrespective of the dependence of the framework's dependence on the mathematical model of the system [20]. In this literature review, DDC is seen as the latter.

1) DATA-DRIVEN MODEL PREDICTIVE CONTROL

MPC is a powerful technique; however, its performance is determined by the accuracy of the representation of the dynamic model used and the assumption that there is no external disturbance. It is not realistic to encapsulate the dynamics of complex nonlinear systems in a model representation and assume that there are no external interactions with the system. Thus, DDMPC are studied, as they use data-driven techniques to extend the MPC frameworks. Historical data from the system is used to model the dynamics of the plant to be used in the MPC framework [4], data-driven approaches have been used to formulate stochastic MPC models to encapsulate the uncertainty that the system endures to autonomously improve the performance of repetitive tasks [1]. If system identification is omitted and the control policy is determined solely from the data or the feedback information, this method is

commonly referred to as data-driven optimal control and can forms a part of model-free frameworks.

Extension of the MPC framework using both unsupervised and supervised learning techniques have been studied. Unsupervised learning techniques include, clustering algorithms [56], mixture Gaussian learning method to detect false data points in a smart grid estimation framework [57], and non-Bayesian learning for fast convergence [58]. Supervised learning techniques utilised include applying regression [59] in conjunction with online modelling methods to estimate the mathematical model of nonlinear time-varying systems. The reader is referred to [60] for the stability analysis of the DDMPC framework.

2) DATA-DRIVEN CONTROLLER TUNING

In control systems with fixed controller architectures, data-driven approaches have been used to fine-tune the controller parameters. Some of the earliest works in this field include the tuning of the PID controller [61]. Prominent iterative methods for controller tuning include IFT and CbT. Non-iterative methods include VRFT and nCBT. These methods are discussed in Section VI.

3) LEARNING-BASED DATA-DRIVEN CONTROL

Adaptive control methods initially were designed for model-based frameworks, which use the plant's dynamical system representation to make decisions whilst handling uncertainties. However, the proposition of model-free learning-based methods for adaptive control was seen as promising as it does not rely on exact physics and mathematical modelling of the considered system. Instead, the aim is to use learning-based methods to iteratively adapt the control law, which better encompasses dealing with the disturbances' negative effects and the effects of parameter variation. As much as this is a method with potential, it comes with its drawbacks of slow convergence and the possibility of not being able to interpret the learned control law [62].

Model-free DDC control methods, which use learning-based methods and data to derive the control law, may be NN based or non-NN based. DLT, LL and ILC [63]–[65] are a non-NN based methods. DLT is a DDC method which is considered a fundamental tool for discrete-time nonlinear systems [63]. LL is classified as a non-NN based machine learning method. ILC is a learning-based method that iteratively updates the control policy for repetitive tasks through successive iterations. Although first proposed in 1978 [64], ILC had not drawn much attention as it was published in Japanese. However, in 1984 [65] the work was published in English. For more details on ILC, the reader is referred to a survey [66] and various industrial applications of ILC [67], [68].

Model-free DDC frameworks have become prevalent amongst the control and robotics communities in the recent past. Particularly NN based learning methods that have been used to develop model-free DDC frameworks, which include RL [7] and learning from demonstration (LfD) [69], [70]. RL approaches have shown the capability to realise the optimal control; common methods or frameworks used are Q-learning and Actor-Critic (AC) architectures. A comprehensive review of some of the earliest works in this field is discussed in [71], [72]. Data-driven policy derivation methods are further discussed in Section VII.

The birth of RL can be attributed to the culmination of trial-and-error search psychology in the animal kingdom, Dynamic Programming (DP) and optimal value functions. DP optimises the input trajectory by using a function where the unknowns are also functions generated by the system's state information in conjunction with a value function [73]. However, the optimisation problem, once reformulated, could potentially be intractable due to the curse of dimensionality. This is a drawback of DP, hence the proposition of Approximate DP (ADP) [74]–[76]. ADP approximates the control policy by using an offline iteration algorithm or an online update algorithm [77]. RL leverages one such ADP method to solve for the optimal policy offline. The design of ADP may take one of many forms that are dependent on the structure of the agent.

RL was formulated with the aim of minimising the loss function over time for dynamical control systems [78], [79]. RL, an area of machine learning, which was developed as an optimal sequential decision-making method, is considered an adaptive control algorithm as it can account for uncertainty without having to be reliant on a finite number of formulated stochastic models like in the DDMPC framework [22], [80]. Unlike MPC and DDMPC frameworks which are reliant on mathematical models, RL is model-independent which is advantageous, particularly for industrial processors that are nonlinear or a MIMO system (possibly both), as it is not a trivial task to model their complexities mathematically [81]. A drawback of MPC is that its performance is proportional to the length of the prediction horizon. However, for more complex systems, to ensure computational feasibility, the prediction horizon is shortened, which could result in sub-optimal results in the long-term [22]. In contrast, some RL algorithms conquer this challenge by pre-computing the optimal solution offline [23], [82]. Furthermore, unlike MPC, RL does not have the online computational demands of trajectory optimisation methods. The development of RL for control systems discussed in [22], [23].

DDC methods have to date been applied to a number of industries and applications, including power electronics [63], [83], [84], data center cooling [85], [86] and UAVs [87]–[89], amongst others.

In summary, this section gives the timeline and development from primitive model-based control techniques to current day model-free control techniques. The taxonomy of this section is summarised in Fig. 3, which classifies methods based on their dependence on a model of the system, the use of data and if the controller tuning methods are iterative or non-iterative.

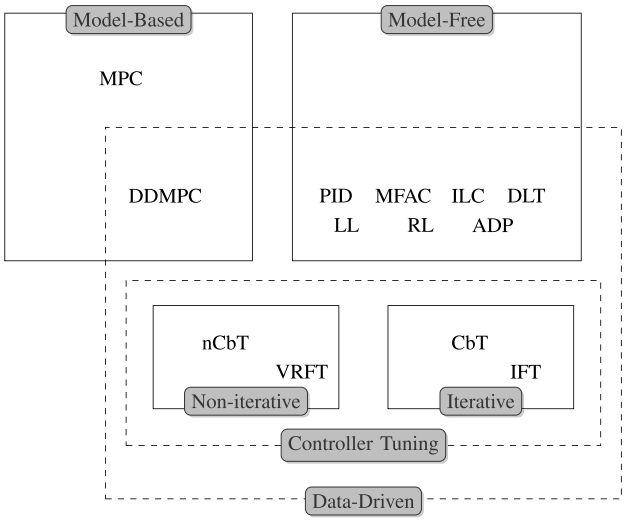


FIGURE 3. The development from model-based to model-free control or both continuous-time and discrete-time methods. The methods are classified based on their dependence on a system model, the use of data and if the controller tuning methods are iterative or non-iterative.

V. MODEL-BASED CONTROL

Model-based control techniques, MPC and its data-driven extension, DDMPC, are discussed in Section V-A and Section V-B, respectively.

A. MODEL PREDICTIVE CONTROL

The main objectives of the MPC controller are to prevent the violation of input and output constraints, maintain outputs within specified boundaries whilst propelling the system to the desired reference trajectory, and control as many process variables as possible with limited available sensors or actuators [90], [91]. The basic structure of the MPC framework is summarised in Fig. 4, and the corresponding MPC trajectory for a SISO system is given in Fig 5. The mathematical model representation, `Dynamic Model`, of the considered process plant, `Plant`, is used in determining the future actuations to be applied to the plant over a prediction horizon, `Predicted Input Control`, and the corresponding predicted observed outputs, `Predicted Output`. From the calculated control actions over the predictive horizon, the first action on the `Predicted Control Input` acts on the dependent variables as a means to account for the changes caused to the system by independent variables. The predictive trajectory may or may not be followed due to disturbances. Independent variables that the controller cannot adjust are taken as disturbances, and dependent variables in these processors are other measurements that represent either control objectives or process constraints. Since the MPC model follows an iterative process, as a result of the inherent nature of feedback algorithms, the output after the first input from the set of actions allocated over the prediction horizon, `Output`, is fed back into the controller through updating the `Dynamic Model` with respect to the reference signal, `Reference`, the objective function, `Objectives`, and constraints, `Constraints`. Based on the residual, the difference between the measured output and the reference set, the prediction horizon is re-initialised, and the next set of control actions are determined. This process is executed multiple times to try and get the system acted upon to behave as desired. Formally, repeatedly solving a constrained optimisation problem to choose the control action whilst accounting for predictions of future costs, disturbances, and constraints over a moving time horizon are known as the RHC. The prediction horizon is iteratively shifted forward, hence MPC is commonly known as the RHC method. The idea of receding horizons dates back to the 1960*s* [42] and was used to ensure constraints are satisfied, limits on control variables and sophisticated feed-forward action are maintained. MPC's predictive capability, ability to optimise over the current horizon while accounting for the future, which is obtained by the iterative optimisation over a finite horizon, and take into account model constraints are some of its many advantages [92]. However, the drawback of MPC includes the computational inefficiencies which arise due to MPC being a complex algorithm. Hence the system dynamics scale [92] and its dependence on the dynamic model of the system. The cost, time and effort of capturing an accurate dynamic model of systems are the largest obstacles in MPC [93], [94].

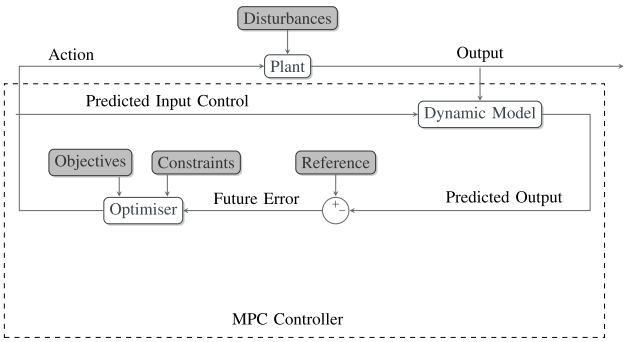


FIGURE 4. Block diagram of MPC architecture showing receding horizon strategy.

The predicted control trajectory in an MPC framework is iteratively updated at each instant t over the interval $[t, t+N]$, where t is the current time, and N is the number of discrete future time-steps which is also known as the prediction horizon length. The corresponding predicted control inputs, $\hat{\mathbf{u}}(t+k|t)$ for $k = 1, \ldots, N$, and outputs, $\hat{\mathbf{y}}(t+k|t)$ for $k = 1, \ldots, N$, are determined based on the plant's dynamic model and the current state $\mathbf{x}_t$. From the set of predicted actuations, only the first actuation is applied to the plant. The plant's state is then re-sampled, and the future predicted trajectory is recalculated [95], [96].

The MPC relies on the discrete-time state-space model of the plant to predict the plant's future actuations over the receding horizon, which is used in the design of the controller and can be expressed by

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \tag{1}$$

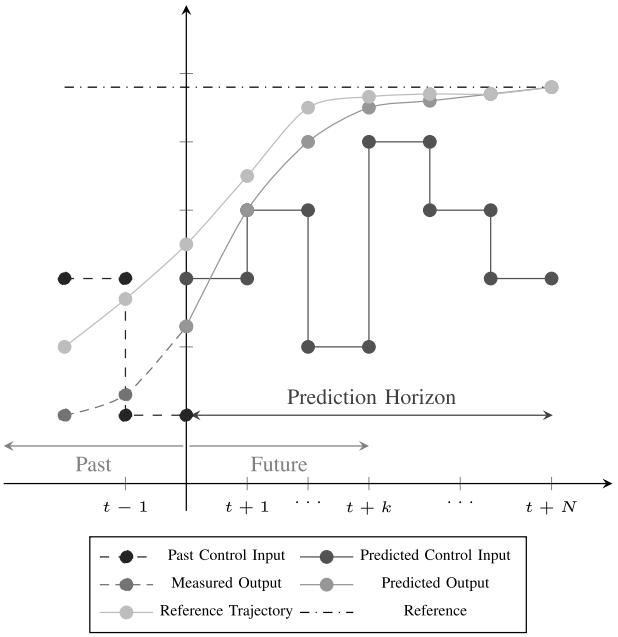


FIGURE 5. Discrete-time trajectory illustrating MPC and receding horizon strategy.

and the corresponding measured output is given by

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t, \tag{2}$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are the discrete state-space plant model dynamic matrices, $\mathbf{u}$ is the control input which is also known as the manipulable variable, $\mathbf{y}$ is the measured output vector, and $\mathbf{x}$ is the state variable vector. The objective is to find the control sequence that minimises the quadratic cost function

$$\begin{aligned} J(\mathbf{u}) &= \sum_{k=1}^{N} \left(\mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right), \\ \text{s.t} \quad \mathbf{Q} &= \mathbf{Q}^T \geq 0, \quad \mathbf{R} = \mathbf{R}^T > 0, \end{aligned} \tag{3}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are respectively the state and the control cost weight matrices. This objective function is subject to the linear inequality constraints on the system inputs:

$$\mathbf{u}_{min} \leq \mathbf{u}_{t+k} \leq \mathbf{u}_{max}, \quad k = 1, \ldots, N, \tag{4}$$

$$\Delta\mathbf{u}_{min} \leq \Delta\mathbf{u}_{t+k} \leq \Delta\mathbf{u}_{max}, \quad k = 1, \ldots, N, \tag{5}$$

where $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$ respectively are the minimum and maximum bounds of the control actions, and $\Delta\mathbf{u}_{min}$ and $\Delta\mathbf{u}_{max}$ are respectively the minimum and maximum control increments.

This general MPC model can be reformulated to be more realistic and include noise or be developed with an infinite prediction horizon or terminal constraint for a more robust model. The reader is referred to [55], [91], [97], for further details on MPC models, to Table 2 for an overview of the literature on the development of MPC and Table 3 for applications of MPC to control problems.

TABLE 2. Overview of MPC literature.

Reference	Approach
[73] (1966)†	DP was developed to solve optimal control.
[43], [98] (1978)†	MPHC or later known as MAC.
[44], [45] (1980)†	DMC.
[99] (1987)†	Generalised Predictive Control (GPC).
[41] (1989)	MPC survey paper.
[100] (2003)	MPC review paper on the application in industry.
[55] (2011)	MPC review paper of the three decades (1980s to 2010).
[101] (2016)	Review and future work on stochastic MPC.

† Key methods used in the development of the MPC framework.

TABLE 3. Overview of MPC applications.

Reference	Application
[102] (2013), [49] (2014) [50] (2017)	Power electronics.
[51] (1999), [52] (2018), [53] (2019)	Data center cooling.
[103] (2012)	Automotive industry.
[54] (2016)	Unmanned autonomous vehicles.
[104] (2017), [105] (2018)	Aerospace systems.
[106] (2018)	Agricultural production

B. DATA-DRIVEN MODEL PREDICTIVE CONTROL

MPC's potential is limited by the accuracy of the model representation of the system and the available actuations. DDMPC is an extension to MPC which aims to provide means to enhance the powerful MPC framework by using data for system identification [1], [107], [108] and encapsulate disturbances in the model of the plant through data-driven stochastic model predictive control to satisfy constraints in the presence of uncertainty and achieve recursive feasibility [1].

The step of model identification estimates the nominal model of the system using data has been prominent for linear systems, but more recently, system identification has been studied for nonlinear systems [109]–[111]. The development of data-driven stochastic MPC, used to encapsulate disturbance in the model, is described by [112]–[115] are summarised by [116].

DDMPC is an adaptive control technique that combines the model-based MPC method with data-driven learning techniques. DDMPC extends on the MPC framework by learning from the trajectory data of the system at every time step to construct a safety set which is used to learn in which region of the state-space the system should operate [4], [70], [116]–[118]. Although DDMPC is an extension of MPC, it shares the drawback of MPC that this framework is dependent on a mathematical model of a system, however, the advantages of DDMPC is that it may be easier to model the system and its uncertainties using data rather than through

merely using physics, and is simpler to integrate into control frameworks than MPC [119].

Applications of the DDMPC framework range from the mechatronics [120] to home assistance appliances [121]. The reader is referred to the following work [116], [122], [123] for details on the guarantees of the robustness of the DDMPC framework. An overview of the literature on DDMPC is given in Table 4, and the applications of DDMPC to control problems are tabulated in Table 5.

TABLE 4. Overview of DDMPC literature.

Reference	Approach
[124] (1987), [125] (1981)†	k-armed bandits. Optimal action; one situation; immediate consequence.
[124] (1987)†	Contextual bandits. Optimal action; many situation; immediate consequence.
[20] (2013)	Survey paper mapping the development from model-based control to DDC.
[112] (2013), [113] (2014), [114] (2015) [115] (2017) [1] (2018)	Stochastic DDMPC uses data in the MPC framework to encapsulate disturbances in the model of the plant to satisfy constraints in the presence of uncertainty and achieve recursive feasibility.
[116] (2018)	Review of data-driven predictive control for autonomous systems. Focuses on MPC with uncertainty and how to use data to efficiently formulate stochastic MPC.
[4] (2018)	Learning MPC for iterative tasks. A DDC framework. Proposes a method to recursively reconstruct the terminal set from current state.
[5], [126] (2020)	Stability and robustness analysis for DDMPC.

† Key methods used in the development of the DDMPC framework.

TABLE 5. Overview of DDMPC applications.

Reference	Application
[93] (2018), [119] (2020), [127] (2021)	Energy management for buildings.
[128] (2021)	Energy management for a semi-closed greenhouse.
[129], [130] (2021)	Data center cooling.
[131] (2021)	Quadcoptor trajectory tracking.
[132] (2019)	Trajectory tracking.

VI. DATA-DRIVEN CONTROLLER TUNING

Data-driven methods used to tune the parameters of fixed controllers include IFT, VRFT, nCbT and CbT.

VRFT and nCbT are offline direct, non-iterative data-driven methods used to optimise the controller. The optimal parameters of the controller are thus identified using a single I/O data set of the control plant. VRFT [133] and nCbT [134] are both used to select the parameters of linear time-invariant systems (LTIs). VRFT formulates the controller tuning problem by introducing a virtual reference signal for parameter identification. However, the nCbT method does not introduce a virtual signal and performs better than VRFT even if the data is noisy as it uses a correlation-based approach.

Given that VRFT and nCbT are offline methods, if any changes are made to the plant, the plant's parameters must be re-tuned. A drawback of VRFT is that its performance relies on whether or not the system dynamics are sufficiently encapsulated in the data set through the plant's sensors. The reader is referred to the following references on the extension of VRFT: applications of VRFT for nonlinear systems [135] which, in contrast to the linear implementation, is an iterative method; an extension of VRFT for MIMO systems [136], and the study of the robustness and other extensions of this method [137].

IFT and CbT are iterative data-driven controller tuning methods. IFT [138] is a model-free method which at each successive iteration, optimises the fixed-structure controller's parameters using the feedback received from the closed-loop system. This technique is suited to doing precise, repetitive tasks. IFT applies the quasi-Newton method, which is a gradient-based method that has its own drawbacks [139], the convergence rate is reliant on how *good* the approximation is of the positive-definite matrix, and the method is computationally demanding with respect to both storage and computation [140]. CbT is a correlation-based tuning method and is closely related to IFT. However, it differs with respect to the means of obtaining the gradient estimates, and CbT only uses one experiment per iteration. The reader is referred to the following references [141], [142] on the extension of CbT to MIMO systems.

A summary of the controller tuning methods are tabulated in Table 7 and the literature on the tuning of controller is tabulated in Table 6.

TABLE 6. Overview of literature on data-driven controller tuning techniques.

Reference	Approach
[138] (1998), [143] (2002)	IFT.
[133] (2000)	VRFT.
[136] (2004), [135] (2006)	Extension of VRT for MIMO and nonlinear systems.
[141] (2004)	CBT.
[137]	Study of robustness on VRFT.
[144] (2012)	nCBT.

VII. MODEL-FREE CONTROL

MFAC [145], as the name suggests, do not require precise quantitative knowledge of the system. This DDC method has been favoured as it simply uses online or offline I/O data measurements of the controlled system to determine the control policy and has the potential to adapt to environmental changes or disturbances [146], without the explicit use of parametric or non-parametric models of the system to be controlled during adaptation [20]. Properties of MFAC include not requiring system identification, controller tuning, controller design specific to the process and an exact mathematical model representing the system's dynamics (including

TABLE 7. Overview of data-driven controller tuning techniques, distinguished based on the use of online and offline data, if the method is iterative or non-iterative, applicable to fixed or adaptive controllers, and if the controller tuning method is suitable for linear or nonlinear systems.

	IFT	CbT	VRFT	nCbT
Online/Offline data	Online	Online	Offline	Offline
Iterative/Non-iterative	Iterative	Non-iterative	Iterative	Non-iterative
Fixed/Adaptive control	Fixed	Fixed	Fixed	Fixed
Linear/Nonlinear	Linear§	Linear§	Linear§	Linear§

§ Extendable to nonlinear systems.

nonlinear dynamics). In addition, closed-loop stability analysis is available to guarantee stability [146].

Given the many advantages of model-free adaptive controllers, the potential of extending their capabilities to various applications in the automatic control industry is currently being studied and applied. An example includes model-free adaptive controllers directly replacing PID controllers used in SISO systems, with the advantage of omitting the step of controller tuning [146]. MFAC framework being applicable to MIMO systems is a characteristic that is both attractive and the reason behind the attention these frameworks are currently receiving.

A summary of the advantages of the data-driven learning methods for policy derivation discussed in this section includes that they do not rely on the exact physics and mathematical modelling of the considered system and can adapt the control law, which better encompasses dealing with the disturbances' negative effects and the effects of parameter variation. However, irrespective of the potential of data-driven control methods for policy derivation, they come with the drawbacks of slow convergence and the possibility of not being able to interpret the learned control law [62].

A comparison table is presented in [20] on the classification of various control methods based on the following characteristics: whether or not either or both online and offline data are used, the system is suitable for SISO or MIMO systems, if the design encapsulates nonlinear model dynamics or only LTI systems, whether or not the optimal policy is iteratively updated or directly learnt from a single data set, if the RL algorithm is an on-policy or off-policy algorithm, whether or not the algorithm is NN based or not, and their respective computational demands.

A particular distinction between model-free control techniques is whether or not the technique is NN based. Non-NN based and NN based techniques are discussed in Section VII-A and Section VII-B, respectively.

A. NON-NEURAL NETWORK BASED METHODS

Prominent non-NN based methods used in policy derivation include DLT, ILC and LL, which are discussed in Section VII-A1, Section VII-A2 and Section VII-A3 respectively.

1) DYNAMIC LINEARISATION TECHNIQUES

Earlier work on MFAC studied the application of DLT for discrete-time SISO nonlinear systems [145], [147]–[150]. Given that this is a model-free framework, a sequence of identical local dynamic linearisation data models were built along the closed-loop system's dynamic operation points using a DLT, with a pseudo-partial derivative (PPD). The I/O measurement data of a controlled plant is used to estimate the time-varying PPD, which is iteratively updated. The DLT includes compact-form dynamic linearisation (CFDL), partial-form dynamic linearisation (PFDL), and full-form dynamic linearisation (FFDL). The reader is referred to [146], [150] for details on these methods, for which stability and convergence can be proven under certain assumptions. Most of these methods have been designed for SISO nonlinear plants; however, they cannot be directly extended and applied to MIMO systems without addressing input coupling. These are discussed in [150]–[152]. These methods are favourable as they do not require external training or testing. However, their computational burden and the impractical assumptions made to prove stability and convergence discourage them from being used.

2) ITERATIVE LEARNING CONTROL

ILC [64], [65] is well-suited for systems that perform repetitive operations through the tracking of output errors and tracks actuations from previous iterations. ILC guarantees convergence as the number of iterations approach infinity. ILC is a model-free data-driven adaptive control method that requires very little knowledge of the plant and uses both online and offline data to directly determine and update the control policy. The reader is referred to the following literature surveys on ILC [67], [153]–[157].

Critically reviewing the ILC method, it is highlighted that the performance of this method with respect to convergence to the desired trajectory relies on unrealistic assumptions, making it an unrealistic method to apply to plants with significant uncertainty [158]–[161].

3) LAZY LEARNING

LL is a class of supervised machine learning algorithms that was applied to the control field [162]. LL was developed to build a relationship between the input and output data. Historical data is used as the training set. In addition, LL algorithms use online data for real-time updating. Examples of LL methods include K-nearest neighbours, local regression and lazy naive Bayes rules. LL is a powerful technique. However, its computational cost is high, a requirement for large amounts of training data, the impact that noisy training

data has on the training phase, and the lack of theoretical analysis are drawbacks [163].

B. NEURAL NETWORK BASED LEARNING

NN parameterised model-free adaptive controllers use NN structures to implicitly represent the system's dynamics. The development of NN based optimal control techniques are commonly classified as RL for optimal control, event-based control, signal processing, machine intelligence for control and intelligent control, amongst others.

NNs are used in MFAC by creating a multilayer perceptron NN with weight factors updated as the controller's behaviour varies. The adaptation of the weighting assists in iteratively reducing the error value. The 'memory' characteristic of the controller is valuable and provides adaptive characteristics which make them suitable for learning-based techniques.

RL, derived from neutral stimulus and reaction, is a machine learning method that envelopes both supervised and unsupervised learning. The increased popularity of RL algorithms is attributed to their success in addressing sequential decision-making problems [17]. RL algorithms aim to develop agents to learn how to take favourable actions in an environment to maximise the notion of cumulative reward. RL methods are particularly used when the state-action space is too large to be completely known but can use some experience samples, or when the model is unknown but experiences can be sampled to determine a policy. RL use NNs to approximate this policy function or a value function.

The three methods used in RL to determine the optimal policies are Dynamic Programming (DP), Monte Carlo (MC) methods and Temporal Difference (TD) methods. From these three methods, DP is mathematically well established but is model-based, MC method is model-free but does not use online data, hence updating the estimate of the value policy happens at the end of the episode [164], [165]. Whilst, the TD method is model-free and is implemented using online data that can be used to update the value function.

1) *Dynamic Programming:* Given that the model precisely encapsulates the plant dynamics, DP can deterministically find the optimal policy, however, it is unrealistic to expect an accurate model of the non-trivial systems. Popular DP methods include policy iteration and value iteration methods [23], [73].
2) *Monte Carlo Method:* MC finds the optimal policy by estimating the average returns for different policies by sampling multiple sequences of states, actions and rewards under the determined policy. MC is most suitable for systems that have finite tasks with explicit terminal states [23], [166].
3) *Temporal Difference:* TD method is widely used in RL as it has a relatively cheap computational cost and can learn from experiences (like MC methods) with bootstrapping (like DP methods). Furthermore, TD is a model-free method and instead learns the dynamics from interactions with the system. Another favourable characteristic of TD is that it does not require waiting until the end of a training episode to update the value function [23], [167].

Adaptive Dynamic Programming (ADP) [74], [75], an extension of DP and an optimal control scheme [168] which is suitable for linear plants with quadratic objective functions over an infinite horizon. This method can be extended to nonlinear plants, models with different cost functions, and systems defined for finite horizons. The reader is referred to the following literature surveys for the development of ADP [169], [170]. The application of NNs to DP problems was proposed to derive the value function, such that the framework is model-free and robust to disturbance. ADP is a TD learning method that updates the current estimate of the value function at either each or over a few iterations rather than at the end of a full episode [167]. This is an attractive characteristic as updates do not only occur at the end of an episode. Some prominent ADP NN based schemes with an adaptive critic structure include Q-learning [171]–[174], SARSA [164] and AC methods.

DP, in a deterministic fashion, finds the optimal policy, however, it is model-based and computationally demanding for complex tasks. Asynchronous or offline DP methods have been developed, however, they perform poorly when less common states are encountered. Both TD and MC approximate DP solutions using less computational power and are model-independent. The MC method finds the optimal value by averaging the value function over the sample trajectories of states, actions and rewards, unfortunately, the variance in the samples trajectories are high. TD combines the ideas of DP and MC methods into one unifying algorithm. TD methods learn from sampled data like in MC methods, while also performing mid-trajectory learning, like in DP, however, TD methods experience high bias due to estimating values through previously estimated values which is commonly referred to as bootstrapping. For a comprehensive introduction to these methods, the reader is referred to [17]. A summary of these methods' characteristics are tabulated in Table 8.

Data-driven optimal control is where RL meets control theory. The controller is designed using input-output data from the system, which is passed through NN based control methods or intelligent methods, commonly referred to as 'black-box' approaches, which implicitly learn the system's dynamics. In contrast to model-based control systems, explainability, robustness and stability provided by deterministic models are not provided or are currently being studied [20].

RL methods commonly model the problem as a Markov Decision Process (MDP). MDP is a multi-stage discrete-time representation of the stochastic optimal control problem and a classical formulation of sequential decision making where both immediate and future rewards are considered [17], [175]. MDPs can be expressed as a tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R}, \gamma \rangle$, where $\mathbf{S}$ is the set of states $\mathbf{s}$, $\mathbf{A}$ is the set of actions $\mathbf{a}$, $\mathbf{P}$ is the set of state transition probabilities $\mathbf{p}$, $\mathbf{R}$ is the set of rewards $\mathbf{r}$, and γ is the discount factor accounting for all rewards, where

TABLE 8. Overview of RL methods, adapted from [23].

Characteristic	Method		
	Dynamic Programming	Monte Carlo	Temporal Difference
Model-based/Model-free	Model-based	Model-free	Model-free
Computational Cost	High	Low	High
Estimate bias	High	Low	High
Estimate variance	Low	High	Low
Value function update	All states simultaneously	After a trajectory	After an experience
Exploration	Exact methods, all states updated	Random initialisation	Performs a random action

$\gamma \in [0, 1]$ [176]. The set of states and actions are specific to time t, hence at any given time t a set of states $\mathbf{s}_t$ is a subset of $\mathbf{S}$ and similarly for actions, state transition probabilities and rewards. The reader is referred to [17], [23] for details on the three different MDPs: fully observable MDPs (FOMDP), partially observable MDPs (POMDP) and semi-MDPs (SMDP).

The RL paradigm, as shown in Fig. 6, consists of two components, the agent and the system. If compared with the closed-loop controller depicted in Fig. 2a, it is noted that the controller is simply replaced with an agent in the RL paradigm. The agent which is the the decision-maker is continuously learning and updating its policy. The agent attempts to learn and conquer the system through meaningful sequential interactions with the system. The system is comprised of everything the agent cannot arbitrarily change. Relating to the overview of process control, Fig. 2a, the agent would be the controller's logic, and everything else would make up the system. RL algorithm's decision-making process is formalised in the MDP.

The optimal solution to a RL problem refers to the policy that generates the highest reward over a trajectory. Formally, the optimal policy must satisfy the principle of optimality which is defined as: the optimal policy π^* is optimal if and only if $V^{\pi *}(\mathbf{s}) \geq V^{\pi \neq \pi^*}(\mathbf{s})$ for all $\mathbf{s} \in \mathbf{S}$ [177].

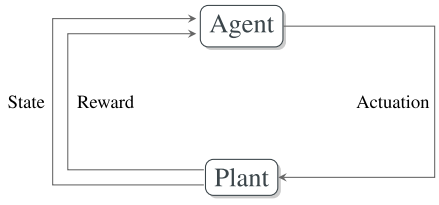


FIGURE 6. RL under MDP framework [17].

Two main model-free methods used in RL algorithms are value-based and policy-based methods. AC approaches are hybrid approach that employs both value functions and policy searches [127].

1) *Value-Based Methods:* Value-based methods do not store an explicit policy but rather a value function from which the policy can be implicitly obtained. The value function V returns the expected value of the return $\mathbf{R}$ of being in an initial state $\mathbf{s}$ and subsequently following the policy π, is defined by the state-value function as follows

$$V^{\pi}(\mathbf{s}) = \mathbb{E}[\mathbf{R}|\mathbf{s}, \pi]. \tag{6}$$

The optimal state-value function is the corresponding state-value function for the optimal policy π^*, defined by

$$V^*(\mathbf{s}) = \max_{\pi} V^{\pi}(\mathbf{s}) \quad \forall\, \mathbf{s} \in \mathbf{S}. \tag{7}$$

Using $V^*(\mathbf{s})$, the optimal policy could be derived by choosing all the actions available at $\mathbf{s}_t$ and selecting the action $\mathbf{a}$ that maximises $\mathbb{E}_{\mathbf{s}_{t+1} \sim \tau(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a})}\left[V^*(\mathbf{s}_{t+1})\right]$. The transition dynamics τ is not available, hence the state-action function is constructed. The state-action function returns the expected value given the initial action $\mathbf{a}$ and the policy π is subsequently followed from the initial state, the state-action value function is defined as

$$\mathbf{Q}^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}[\mathbf{R}|\mathbf{s}, \mathbf{a}, \pi]. \tag{8}$$

Given the state-action value function $\mathbf{Q}^{\pi}(\mathbf{s}, \mathbf{a})$, the optimal policy can be retrieved by greedily choosing the action with the highest value. Under this policy, the value function can be defined by maximising $\mathbf{Q}^{\pi}(\mathbf{s}, \mathbf{a})$: $V^*(\mathbf{s}) = \max_{\mathbf{a}} \mathbf{Q}^{\pi}(\mathbf{s}, \mathbf{a})$ [178].

Prominent value-based methods are SARSA and Q-learning. Value-based methods are best suited for when using a finite set of actions, rather than continuous action space problems.

2) *Policy-Based Methods:* Policy-based methods directly learn the optimal control policy π^* and do not need to maintain a value function model. Frequently, a parameterised policy with respect to θ, π_θ is chosen. The parameter are selected to maximise the expected return $\mathbb{E}[\mathbf{R}|\theta]$ using either gradient-based or gradient-free optimisation [178]. Successfully trained NNs with encoded policies are discussed for both gradient-based methods in [179] and gradient-free methods in [180]. Policy-based methods are discussed in detail by [181]. Policy-based methods are useful when the action space is continuous or stochastic. One disadvantage of policy-based methods is that they use the MC technique, which uses the total rewards. As a result, the agent has to traverse an entire episode before any learning occurs,

which potentially results in a high variance when there are drastic changes.

3) *Actor-Critic Methods:* AC method, shown in Fig. 7, combines the benefits of learned value functions and policy search methods. The AC methods are TD methods with two independent memory structures representing the policy and the value function. The actor-network determines how the agent behaves (policy-based) by proposing a set of possible actions given a state. The critic-network measures how *good* the action taken is (value-based) and returns the probability distribution over the actions that an agent can take based on the given state. AC methods are TD learning methods that do not use the total reward. Instead, a critic model approximates the value function at each discrete time-step, unlike policy-based methods based on MC, which increases the learning rate. The values function replaces the reward function of a policy gradient algorithm that calculates rewards at the end of the episode [178] and instead updates the value function within the episode.
AC is an on-policy method with two separate parametric structures represented by NNs, the actor-network for optimal policy evaluation and the critic-network for the value function. The actions taken by the agent or the actor-network are evaluated by the critic, which represents the reward function, and the objective function using the TD approach [165].

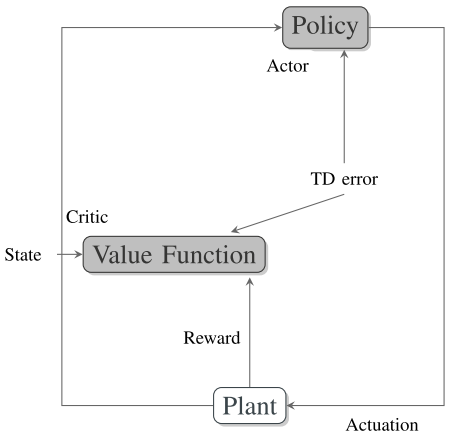


FIGURE 7. Actor-critic structure based on [165].

Q-learning and AC methods are prominently used methods in data-driven learning-based MFAC control systems. Q-learning [182] is a RL method that aims to learn the value of applying an action in a particular state. Q-learning is particularly considered an adaptive control method based on its inherent properties of stochastic transitions and rewards without adaptation. Q-learning is formulated as a finite-state, finite-action MDP, which derives the optimal policy by maximising the expected value of the total reward over a series of successive iterations. Q-learning and Deep Q Network (DQN) are off-policy methods with a slow convergence rate but high efficiency. Unlike the value-based method, Q-learning and AC methods guarantees convergence for nonlinear methods, have a reduced variance estimate of the expected value, and their sampling is efficient via the TD updates [165].

RL algorithms use the three aforementioned methods, DP, MC and TD, to solve for the optimal policy coupled with value-based, policy-based and AC methods. DQN [183], Deterministic Policy Gradient (DPG) [184], Deep DPG (DDPG) [185], Trust Region Policy Optimisation (TRPO) [186] and Proximal Policy Optimisation (PPO) [187] are major contributions to the field of RL and have been widely applied to control systems in determining the optimal policy. A summary of the characteristics of these RL methods is described in Table 10. The reader is referred to the following reviews on RL methods [19], [22], [23], [188].

A summary of the properties of the MPC method and an array of the data-driven optimal control methods are tabulated in Table 9, a summary of notable related literature on DDC for MFAC is given in Table 11, and Table 12 tabulates the learning-based data-driven applications.

VIII. EMERGING TRENDS

The ultimate goal of automated control would be to develop a uniform data-driven framework that is based solely on the I/O measurements and is widely applicable to various industries. RL and deep RL do hold promise in this regard; however, they are still in their infancy to obtain this for complex systems.

In this survey, the use of data in control frameworks is studied. From the literature, it is seen that model-based techniques have been extensively studied and applied. However, model-free methods are still in their embryonic stage, especially given the limited theoretical analysis. More recently, complementary model-based and data-driven control frameworks like DDMPC, data used in the study of controllers, system identification, and uncertainty modelling have been prominent over modular methods. The gap in the literature is the application and development of multi-scale and hierarchical learning structures, such as using learning methods alongside model-based controllers or pre-processed offline data, which could be used in feature extraction. Furthermore, the literature on the handling of uncertainty does not account for irregularities such as time delays or feedback over varying time intervals but only noise in measurement.

It is highlighted that the development of the theoretical analysis of model-free methods has not been established. Stability, robustness and convergence guarantees are nascent properties in process control. However, proving stability for nonlinear systems and for model-free frameworks [7] is not trivial. Challenges to prove stability and convergence under stochastic conditions include proving effectiveness in terms of performance, learning rate and utilised reward function [17], [164]. This is one of the main challenges with RL.

TABLE 9. Overview of the characteristics of control methods. The methods are distinguished based on their dependence on a dynamic model of a system, the use of online or offline data, in what regard is the method using the feedback information, whether the controller is fixed or adaptive and if the control method is suitable for linear or nonlinear systems.

	PID	MPC	DDMPC	ILC	MFAC	ADP	LL
Model-free/Model-Based	Model-free	Model-based	Model-based	Model-free	Model-free	Model-free	Model-free
Online/offline data	Online	Online	Online	Online and Offline	Online	Online	Online
Data-driven	Feedback signal	Feedback signal	Modelling uncertainty	Learn control policy	Learn control policy	Learn control policy	Learn control policy
Iterative/Non-iterative	Iterative	Iterative	Iterative	Iterative	Iterative	Iterative	Iterative
Fixed/adaptive control	Fixed	Adaptive	Adaptive	Adaptive	Adaptive	Adaptive	Adaptive
Linear/nonlinear	Linear	Linear§	Linear§	Nonlinear	Nonlinear	Nonlinear	Nonlinear

§ Extendable to nonlinear systems.

TABLE 10. Overview of RL algorithms used for control systems.

	Method				
Characteristic	**DQN**	**DPG**	**DDPG**	**TRPO**	**PPO**
RL Method	TD	MC	-	-	-
State	Continuous	Continuous	Continuous	Continuous	Continuous
Action	Discrete	Continuous	Continuous	Continuous	Continuous
On/off policy	Off-policy	Off-policy	Off-policy	On-policy	On-policy

TABLE 11. Overview of literature on DDC for MFAC.

References	Approach
[175] (1957)†	Markov Decision Process (MDP).
[64] (1978)‡, [65] (1984)‡	ILC.
[189] (1994), [190] (2018)	MFAC.
[191] (1991)‡‡, [192] (1995)‡‡	Primitive work and overview of NN based work for optimal control.
[193] (1998)‡	Simultaneous Perturbation Stochastic Approximation (SPSA) based model-free control.
[194] (2003)‡, [195] (2007)‡	Extremum seeking methods.
[150] (2011)‡	DLT, PPD, CFDL and PFDL.
[196] (2013)‡‡	A survey of RL in robotics.
[197] (2015)	An overview of data-based techniques focused on modern industry.
[183] (2013)†	DQN.
[184] (2014)†	DPG.
[185] (2015)†	DDPG.
[186] (2015)†	TRPO.
[187] (2017)†	PPO.
[7] (2018)‡‡	Analysis of using RL for control.
[19] (2018)	An overview of model-based and data-driven adaptive control.
[198] (2013)‡‡, [199] (2016)‡‡, [200] (2017)‡‡	RL and machine intelligence reviews for optimal control.
[201] (2020)‡‡	A survey on the recent advances in robot LfD.
[198] (2013)‡‡, [199] (2016)‡‡, [200] (2017)‡‡	RL and machine intelligence reviews for optimal control.

† Key methods used in the development of MFAC framework.
‡ Non-NN based.
‡‡ NN based.

TABLE 12. Overview of model-free learning-based data-driven applications.

Reference	Application
[83] (2017), [202] (2021)	Power electronics.
[203], [204] (2019)	Energy management.
[205] (2019)	Unmanned autonomous vehicle.
[206] (2021)	Autonomous vehicles.
[85], [207] (2021)	Data center cooling.

Recent works on theoretical analysis the formalisation and analysis include [5], [126], [126].

Several other areas of improvement of RL methods include accounting for data inefficiency, constraint handling, means to discourage policies from arriving at intractable states, and the construction of representative simulators. Data inefficiency refers to the requirement of lengthy periods of training data to improve the efficiency of a policy derivation and initial agent training, especially if simulators cannot be used in the training process due to their inaccuracies. Emerging fields used to try and inject prior knowledge into the agent include transfer learning [207], [208], including the concept of a replay buffer or experience replay [183], [209], [210] as used by DQN, and increasing learning efficiency using eligibility traces which essentially combines TD and MC methods into unifying algorithm which allows for agents to update multiple value functions per iteration, like MC, without termination of an episode [17]. Alternative methods to increase the rate of the training process includes exploiting heuristics for RL, such as heuristically accelerated RL (HARL) [211], [212] and meta RL [213]–[215] which use simulations to train the agent; RNN is a common algorithm used in this regard. Finally, alternative methods suggest using two modular structures instead, one for offline decision making and another for online high-level RL.

Another critical challenge of using RL in process control is scalability. Emerging trends include using multi-agent RL

methods [216] and LfD [4]. Since exact methods are not feasible for problems with more than 100 states [23], [183], [186], [210], recent work with numerous states have used multi-agent RL to achieve optimality [217]. Q-learning and other deep RL methods have been useful for various industrial applications.

The promise of RL agents in a plethora of industries can be unlocked with the development of robust and *good* agents. The objective of RL algorithms is to develop an agent to take actions to maximise the cumulative reward. The training process of these agents requires a high volume of trial-and-error episodes in a given environment to optimise for the given reward function. In the light of safety and being cost-conscious, high fidelity simulators are nascent, especially to derive research on the development of RL-based algorithms [218]–[222]. With the increase in computing power and the availability of vast amounts of data developing simulators that apply a mathematical function to input data and returns an output is possible. Some commonly used simulators include MATLAB Simulink and ANSYS for engineering problems, Gazebo and MuJoCo for robotics and physics-based simulations, Bottleneck simulators which are model-based RL simulators that have also been proposed [223], amongst others.

Extensions to simulators include digital twins [224]–[226]. Digital twins provide a virtual representation of the real-time digital counterpart of physical systems or processors. A digital thread is a data pipeline used to obtain data through sensors from the design stage to build and, finally, the operation of the physical system or end product. This obtained data is then feedback to the digital twin. Using the amalgamation of the information from the digital thread with the digital twin, performance information can be extracted, and credible updates made can then be applied along the design, production, and end product or system stages. Thus, a means to holistically optimise the end-to-end process. Both manufacturing and engineering industries are moving from using knowledge-based intelligent processes to data-driven, and knowledge-enabled smart processes [227], [228]. The former has been used for informed decision making, whilst the latter uses real-time transmission and analysis of data across the end-to-end process with the aid of simulators and optimisation mechanisms, providing positive impacts throughout the process. These techniques are used to improve the performance of end-to-end cycles of engineering or manufacturing but also are suggested to be used to build resilient models by incorporating preventative measures that account for disruption risks. These frameworks make use of cyber-physical integration and digital twins. The reader is referred to [208] for manufacturing applications using digital twins and cyber-physical systems, [229] for the discussion of managing disruption risks, and [230] for a survey on digital twins technologies, techniques and engineering perspectives.

Through the evolution of controller frameworks, NN-based techniques have particularly been prominent for MFAC. These NN-based control policy derivation techniques have been critically discussed in this review. Their black-box nature, in most cases, provides an improvement to the control method and thereby, the system performance, however, they lack in providing insight into the updates and development made through the stages of training and adaptation. The need for white-box models or techniques which are explainable and interpretable in both design and inner logic is crucial to unleashing further enhancements in controller designs to make context-based recommendations and to increase user trust through transparency [231]–[235]. This area of research is commonly referred to as Explainable Artificial Intelligence (XAI).

A summary of references related to emerging trends are given in Table 13.

TABLE 13. Overview of literature on emerging trends.

References	Approach
[7], [237] (2019), [5] (2020), [126], [126] (2021)	Stability and robustness analysis for data-driven optimal control techniques.
[209] (2018), [238] (2020)	A survey on transfer learning in deep RL.
[239] (2020)	Data-driven hierarchical predictive learning.
[240] (2016), [225] (2018), [226] (2019), [227] (2020)	Digital twin.
[228], [231] (2019) [230] (2020)	Applications and survey of digital twin.
[232] (2018), [233] (2018), [234] (2020), [235] (2020), [236] (2021)	Interpretable and explainable black-box techniques and XAI.

IX. CONCLUSION

The development of model-based predictive control to data-driven control techniques is motivated by eliminating the step of mathematically modelling plants, especially nonlinear complex ones, to develop policies robust to disturbances directly from I/O data and to use data to fine-tune fixed design model-based controllers.

It is highlighted that model-based frameworks are restricted to the accuracy of the mathematical model representing the plant. However, if the model accurately represents the plant, the model-based framework with fixed controllers is robust. The paradigm of learning the control policy directly from the feedback signal has been prominent in the recent past as it discards the requirement of modelling the physics of the plant but, as a result, has to explore a greater search space in deriving the optimal policy.

In this review, the taxonomy and timeline of data-driven control techniques were given, the corresponding references have been summarised in the respective sections. It is noted that there is an overlap of studies between the control and the RL communities working on developing robust adaptive optimal policies using online I/O data from the controlled plant. Drawbacks of these methods include having to optimise weight functions, parameters and other coefficients of

the learning functions to improve the performance of these methods. These methods are powerful and hold promise, but their potential is restricted by the limited theoretical analysis of convergence, stability and robustness.

Future research in this field would focus on providing the theoretical analysis for the RL based methods, constructing high fidelity simulators which in turn would be a catalyst in the development and research in this field, providing insight to learning-based black-box techniques such that they are interpretable and explainable, commonly referred to as XAI, as well as optimising the end-to-end process of developing and actualising control frameworks with the aid of digital twins and digital threads. Thus, with the ultimate goal of developing a uniform framework that can be used for adaptive optimal control across various applications; a framework that is independent of controller tuning and system identification.

REFERENCES

[1] U. Rosolia, X. Zhang, and F. Borrelli, "Data-driven predictive control for autonomous systems," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 259–286, May 2018.

[2] P. J. Antsaklis and A. Rahnama, "Control and machine intelligence for system autonomy," *J. Intell. Robotic Syst.*, vol. 91, no. 1, pp. 23–34, Jul. 2018.

[3] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 3, no. 1, pp. 269–296, May 2020.

[4] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. A data-driven control framework," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1883–1896, Jul. 2018.

[5] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 909–924, Mar. 2020.

[6] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," 2019, *arXiv:1906.04679*.

[7] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annu. Rev. Control*, vol. 46, pp. 8–28, Jan. 2018.

[8] J. Dorsey, *Continuous and Discrete Control Systems*. Publishing House of Electronics Industry, 2002.

[9] R. Burns, *Advanced Control Engineering*. Amsterdam, The Netherlands: Elsevier, 2001.

[10] D. Nelson, *The Penguin Dictionary of Mathematics*. U.K.: Penguin, 2008.

[11] M. Fliess and S. T. Glad, "An algebraic approach to linear and nonlinear control," in *Essays on Control: Perspectives in the Theory and Its Applications*, H. L. Trentelman and J. C. Willems, Eds. Boston, MA, USA: Springer, 1993, pp. 223–267, doi: 10.1007/978-1-4612-0313-1_8.

[12] O. Yaniv, *Quantitative Feedback Design of Linear and Nonlinear Control Systems*. Springer, 1999.

[13] G. C. Runger and T. R. Willemain, "Model-based and model-free control of autocorrelated processes," *J. Quality Technol.*, vol. 27, no. 4, pp. 283–292, Oct. 1995.

[14] K. J. Astrom. *Control Sysytem Design*. Accessed: Feb. 2, 2022. [Online]. Available: https://www.cds.caltech.edu/murray/courses/cds101/fa02/caltech/astrom-ch1.pdf

[15] K. J. Astrom and T. Hägglund, "Advanced PID control," *IEEE Control Syst.*, vol. 26, no. 1, pp. 98–101, Feb. 2006.

[16] M. Athans and P. L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*. Chelmsford, MA, USA: Courier Corporation, 2013.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[18] C. Okoli, "A guide to conducting a standalone systematic literature review," *Commun. Assoc. Inf. Syst.*, vol. 37, no. 1, p. 43, Nov. 2015.

[19] M. Benosman, "Model-based vs data-driven adaptive control: An overview," *Int. J. Adapt. Control Signal Process.*, vol. 32, no. 5, pp. 753–776, May 2018.

[20] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Inf. Sci.*, vol. 235, pp. 3–35, Jun. 2013. 10.1016/j.ins.2012.07.014

[21] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Event-triggered control for discrete-time systems," in *Proc. Amer. control Conf.*, Jun. 2010, pp. 4719–4724.

[22] J. Shin, T. A. Badgwell, K.-H. Liu, and J. H. Lee, "Reinforcement learning—Overview of recent progress and implications for process control," *Comput. Chem. Eng.*, vol. 127, pp. 282–294, Aug. 2019, doi: 10.1016/j.compchemeng.2019.05.029.

[23] R. Nian, J. Liu, and B. Huang, "A review on reinforcement learning: Introduction and applications in industrial process control," *Comput. Chem. Eng.*, vol. 139, Aug. 2020, Art. no. 106886, doi: 10.1016/j.compchemeng.2020.106886.

[24] J. C. Maxwell, "On governors," *Proc. Roy. Soc. London*, vol. 16, pp. 270–283, 1867. [Online]. Available: http://www.jstor.org/stable/112510

[25] S. Bennett, "A brief history of automatic control," *IEEE Control Syst.*, vol. 16, no. 3, pp. 17–25, Jun. 1996, doi: 10.1109/37.506394.

[26] A. E. Bryson, "Optimal control-1950 to 1985," *IEEE Control Syst.*, vol. 16, no. 3, pp. 26–33, Jun. 1996.

[27] N. Wiener, "Generalized harmonic analysis," *Acta Math.*, vol. 55, no. 1, pp. 117–258, Dec. 1930.

[28] S. Bennett, *A History of Control Engineering*. no. 47. Edison, NJ, USA: IET, 1993, pp. 1930–1955.

[29] N. Andrei, "Modern control theory," *Stud. Informat. Control*, vol. 15, no. 1, p. 51, 2006.

[30] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, Apr. 1988.

[31] L. A. Zadeh, "Fuzzy logic = computing with words," in *Computing With Words in Information/Intelligent Systems 1*. Springer, 1996, pp. 3–23.

[32] G. Welch *et al.*, *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA, 1995.

[33] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[34] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh, "Bayesian optimization for adaptive experimental design: A review," *IEEE Access*, vol. 8, pp. 13937–13948, 2020.

[35] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, Oct. 2006.

[36] P. M. Larsen, "Industrial applications of fuzzy logic control," *Int. J. Man-Mach. Stud.*, vol. 12, no. 1, pp. 3–10, 1980.

[37] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, part II," *IEEE Trans. Syst., Man Cybern.*, vol. 20, no. 2, pp. 419–435, Mar. 1990.

[38] D. Arcos-Aviles, J. Pascual, F. Guinjoan, L. Marroyo, G. García-Gutiérrez, R. Gordillo-Orquera, J. Llanos-Proaño, P. Sanchis, and T. E. Motoasca, "An energy management system design using fuzzy logic control: Smoothing the grid power profile of a residential electro-thermal microgrid," *IEEE Access*, vol. 9, pp. 25172–25188, 2021.

[39] K. Thenmalar, "Fuzzy logic based load frequency control of power system," *Mater. Today, Proc.*, vol. 45, pp. 8170–8175, Jan. 2021.

[40] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura, "Industrial applications of the Kalman filter: A review," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5458–5471, Jan. 2013.

[41] E. C. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice-A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.

[42] H. Hermes, "Foundations of optimal control theory," *IEEE Trans. Autom. Control*, vol. AC-13, no. 2, pp. 222–223, Apr. 1968.

[43] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control. Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[44] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control? A computer control algorithm," in *Proc. Joint Autom. control Conf.*, no. 17, 1980, p. 72.

[45] D. M. Prett and R. Gillette, "Optimization and constrained multivariable control of a catalytic cracking unit," in *Proc. Joint Autom. control Conf.*, no. 17, 1980, p. 73.

[46] G. Pannocchia, J. B. Rawlings, and S. J. Wright, "Fast, large-scale model predictive control by partial enumeration," *Automatica*, vol. 43, no. 5, pp. 852–860, May 2007.

[47] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J Robust Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2007.

[48] Y. Wang and S. Boyd, "Online Optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Jun. 2010.
[49] S. Vazquez, J. Leon, L. Franquelo, J. Rodriguez, H. A. Young, A. Marquez, and P. Zanchetta, "Model predictive control: A review of its applications in power electronics," *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 16–31, Mar. 2014.
[50] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 935–947, Nov. 2017.
[51] L. Ljung, *System Identification-Theory for the User*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
[52] N. Lazic, T. Lu, C. Boutilier, M. Ryu, E. Wong, B. Roy, and G. Imwalle, "Data center cooling using model-predictive control," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–10. [Online]. Available: https://pub-tools-public-publication-data.storage.googleapis.com/pdf/c9fbf966bb9f552def1528116e691eb9399c89c0.pdf
[53] S. S. Gill, P. Garraghan, V. Stankovski, G. Casale, R. K. Thulasiram, S. K. Ghosh, K. Ramamohanarao, and R. Buyya, "Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge," *J. Syst. Softw.*, vol. 155, pp. 104–129, Sep. 2019, doi: 10.1016/j.jss.2019.05.025.
[54] J. Dentler, S. Kannan, M. A. O. Mendez, and H. Voos, "A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors," in *Proc. IEEE Conf. Control Appl. (CCA)*, Sep. 2016, pp. 519–525.
[55] J. H. Lee, "Model predictive control: Review of the three decades of development," *Int. J. Control, Autom. Syst.*, vol. 9, no. 3, pp. 415–424, Jun. 2011.
[56] G. Goebel and F. Allgöwer, "Semi-explicit MPC based on subspace clustering," *Automatica*, vol. 83, pp. 309–316, Sep. 2017.
[57] A. Foroutan and F. Salmasi, "Detection of false data injection attacks against state estimation in smart grids based on a mixture Gaussian distribution learning method," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 2, no. 4, pp. 161–171, Jul. 2017.
[58] A. Nedić, A. Olshevsky, and C. A. Uribe, "Fast convergence rates for distributed non-Bayesian learning," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5538–5553, Nov. 2017.
[59] J. Hu, M. Zhou, X. Li, and Z. Xu, "Online model regression for nonlinear time-varying manufacturing systems," *Automatica*, vol. 78, pp. 163–173, Apr. 2017.
[60] J. Bongard, J. Berberich, J. Köhler, and F. Allgöwer, "Robust stability analysis of a simple data-driven model predictive control approach," 2021, *arXiv:2103.00851*.
[61] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *J. Dyn. Syst., Meas., Control*, vol. 115, no. 2B, pp. 220–222, Jun. 1993.
[62] S. P. Nageshrao, G. A. D. Lopes, D. Jeltsema, and R. Babuska, "Port-Hamiltonian systems in adaptive and learning control: A survey," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1223–1238, May 2016.
[63] Z. Hou, R. Chi, and H. Gao, "An overview of dynamic-linearization-based data-driven control and applications," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4076–4090, May 2017.
[64] M. Uchiyama, "Formation of high-speed motion pattern of a mechanical arm by trial," *Trans. Soc. Instrum. Control Eng.*, vol. 14, no. 6, pp. 706–712, 1978.
[65] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984.
[66] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, May 2006.
[67] Y. Chen and C. Wen, *Iterative Learning Control: Convergence, Robustness and Applications*. London, U.K.: Springer, 1999.
[68] J. H. Lee and K. S. Lee, "Iterative learning control applied to batch processes: An overview," *Control Eng. Pract.*, vol. 15, no. 10, pp. 1306–1318, 2007.
[69] S. Schaal, "Learning from demonstration," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 1040–1046.
[70] U. Rosolia, X. Zhang, and F. Borrelli, "Robust learning model predictive control for iterative tasks: Learning from experience," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 1157–1162.
[71] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Syst.*, vol. 12, no. 2, pp. 19–22, Apr. 1992.
[72] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artif. Intell.*, vol. 72, nos. 1–2, pp. 81–138, Jan. 1995.
[73] R. Bellman, "Dynamic programming," *Science*, vol. 153, nos. 37–31, pp. 34–37, 1966.
[74] P. J. Werbos, W. Miller, and R. Sutton, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*, vol. 3. Cambridge, MA, USA: MIT Press, 1990, pp. 67–95.
[75] P. Werbos, "Approximate dynamic programming for realtime control and neural modelling," in *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. New York, NY, USA: Van Nostrand Reinhold, 1992, pp. 493–525.
[76] M. R. K. Mes and A. P. Rivera, "Approximate dynamic programming by practical examples," in *Markov Decision Processes in Practice*. Springer, 2017, pp. 63–101.
[77] H.-G. Zhang, X. Zhang, Y.-H. Luo, and J. Yang, "An overview of research on adaptive dynamic programming," *Acta Automat. Sinica*, vol. 39, no. 4, pp. 303–311, Apr. 2013.
[78] J. Rawlings and D. Mayne, "Postface to model predictive control: Theory and design," *Nob Hill Pub*, vol. 5, pp. 155–158, 2012.
[79] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, vol. 2. Madison, WI, USA: Nob Hill Publishing, 2017.
[80] K. Rawlik, M. Toussaint, and S. Vijayakumar, "On stochastic optimal control and reinforcement learning by approximate inference," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1–16.
[81] C. T. Maravelias and C. Sung, "Integration of production planning and scheduling: Overview, challenges and opportunities," *Comput. Chem. Eng.*, vol. 33, no. 12, pp. 1919–1930, Dec. 2009.
[82] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.
[83] H. Zhang, J. Zhou, Q. Sun, J. Guerrero, and D. Ma, "Data-driven control for interlinked AC/DC microgrids via model-free adaptive control and dual-droop control," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 557–571, Mar. 2017.
[84] K. Prag, M. Woolway, and T. Celik, "Data-driven model predictive control of DC-to-DC buck-boost converter," *IEEE Access*, vol. 9, pp. 101902–101915, 2021.
[85] Q. Zhang, Z. Meng, X. Hong, Y. Zhan, J. Liu, J. Dong, T. Bai, J. Niu, and M. J. Deen, "A survey on data center cooling systems: Technology, power consumption modeling and control strategy optimization," *J. Syst. Archit.*, vol. 119, Oct. 2021, Art. no. 102253.
[86] Y. Liu, D. V. Le, and R. Tan, "A data-assisted first-principle approach to modeling server outlet temperature in air free-cooled data centers," *Future Gener. Comput. Syst.*, vol. 129, pp. 225–235, Apr. 2022.
[87] M. B. Vankadari, K. Das, C. Shinde, and S. Kumar, "A reinforcement learning approach for autonomous control and landing of a quadrotor," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2018, pp. 676–683.
[88] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.
[89] V. J. Hodge, R. Hawkins, and R. Alexander, "Deep reinforcement learning for drone navigation using sensor data," *Neural Comput. Appl.*, vol. 33, no. 6, pp. 2015–2033, Mar. 2021.
[90] S. J. Qin and T. A. Badgwell, "IDGTE social visit," *Power Eng.*, vol. 5, no. 4, p. 41, 2001.
[91] Seborg. (2003). *Chapter 20 Model Predictive Control 20.1 OVERVIEW OF MODEL PREDICTIVE CONTROL.* Process Dynamics and Control. [Online]. Available: http://folk.ntnu.no/skoge/vgprosessregulering/papers-pensum/seborg-c20ModelPredictiveControl.pdf
[92] C. Ekaputri and A. Syaichu-Rohman, "Model predictive control (MPC) design and implementation using algorithm-3 on board SPARTAN 6 FPGA SP605 evaluation kit," in *Proc. 3rd Int. Conf. Instrum. Control Autom. (ICA)*, Aug. 2013, pp. 115–120.
[93] F. Smarra, A. Jain, T. de Rubeis, D. Ambrosini, A. D'Innocenzo, and R. Mangharam, "Data-driven model predictive control using random forests for building energy optimization and climate control," *Appl. Energy*, vol. 226, pp. 1252–1272, Sep. 2018.
[94] J. Wang, S. Li, H. Chen, Y. Yuan, and Y. Huang, "Data-driven model predictive control for building climate control: Three case studies on different buildings," *Building Environ.*, vol. 160, Aug. 2019, Art. no. 106204, doi: 10.1016/j.buildenv.2019.106204.

[95] J. E. Normey-Rico and E. F. Camacho, *Control of Dead-Time Processes*, vol. 462. Springer, 2007.
[96] S. V. Raković and W. S. Levine, *Handbook of Model Predictive Control*. Springer, 2018.
[97] B. Kouvaritakis, *Model Predictive Control: Classical, Robust, Stochastic*, vol. 36. Bookshelf, 2016. no. 6.
[98] J. Richalet, "Algorithmic control of industrial processes," in *Proc. 4th IFAC Symp. Identification System Parameter Estimation*, 1976, pp. 1119–1167.
[99] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive Control—Part II extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, Mar. 1987.
[100] S. J. Qin and T. A. Badgwell, "Process control dynamic," *Control Eng. Pract.*, vol. 11, pp. 733–764, Oct. 2003.
[101] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Syst. Mag.*, vol. 36, no. 6, pp. 30–44, Dec. 2016.
[102] J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. A. Young, and C. A. Rojas, "State of the art of finite control set model predictive control in power electronics," *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp. 1003–1016, May 2013.
[103] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *Proc. IEEE Int. Conf. Control Appl.*, Oct. 2012, pp. 295–302.
[104] U. Eren, A. Prach, B. B. Koçer, and S. V. Raković, E. Kayacan, and B. Açikmeşe, "Model predictive control in aerospace systems: Current state and opportunities," *J. Guid., Control, Dyn.*, vol. 40, no. 7, pp. 1541–1566, Mar. 2017.
[105] S. Di Cairano and I. V. Kolmanovsky, "Real-time optimization and model predictive control for aerospace and automotive applications," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 2392–2409.
[106] Y. Ding, L. Wang, Y. Li, and D. Li, "Model predictive control and its application in agriculture: A review," *Comput. Electron. Agricult.*, vol. 151, pp. 104–117, Aug. 2018.
[107] L. Ljung, "Perspectives on system identification," *Annu. Rev. Control*, vol. 34, no. 1, pp. 1–12, 2010.
[108] W. Edwards, G. Tang, G. Mamakoukas, T. Murphey, and K. Hauser, "Automatic tuning for data-driven model predictive control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 7379–7385.
[109] X. Ge, Z. Luo, Y. Ma, H. Liu, and Y. Zhu, "A novel data-driven model based parameter estimation of nonlinear systems," *J. Sound Vibrat.*, vol. 453, pp. 188–200, Aug. 2019.
[110] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte, "Data-driven approximation of the Koopman generator: Model reduction, system identification, and control," *Phys. D: Nonlinear Phenomena*, vol. 406, May 2020, Art. no. 132416.
[111] Y. Li and J. Duan, "A data-driven approach for discovering stochastic dynamical systems with non-Gaussian Lévy noise," *Phys. D, Nonlinear Phenomena*, vol. 417, Mar. 2021, Art. no. 132830.
[112] G. C. Calafiore and L. Fagiano, "Stochastic model predictive control of LPV systems via scenario optimization," *Automatica*, vol. 49, no. 6, pp. 1861–1866, 2013.
[113] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.
[114] M. P. Vitus, Z. Zhou, and C. J. Tomlin, "Stochastic control with uncertain parameters via chance constrained control," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2892–2905, Oct. 2016.
[115] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Stochastic MPC with offline uncertainty sampling," *Automatica*, vol. 81, no. 1, pp. 176–183, 2017.
[116] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning model predictive control," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 5115–5120.
[117] M. Brunner, U. Rosolia, J. Gonzales, and F. Borrelli, "Repetitive learning model predictive control: An autonomous racing example," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 2545–2550.
[118] U. Rosolia, X. Zhang, and F. Borrelli, "A stochastic MPC approach with application to iterative learning," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 5152–5157.
[119] A. Kathirgamanathan, M. De Rosa, E. Mangina, and D. P. Finn. (2021). *Data-Driven Predictive Control for Unlocking Building Energy Flexibility: A Review*. [Online]. Available: https://creativecommons.org/licenses/by/4.0/
[120] A. Norouzi, H. Heidarifar, M. Shahbakhti, C. R. Koch, and H. Borhan, "Model predictive control of internal combustion engines: A review and future directions," *Energies*, vol. 14, no. 19, p. 6251, Oct. 2021.
[121] L. Tagliavini, A. Botta, P. Cavallone, L. Carbonari, and G. Quaglia, "On the suspension design of paquitop, a novel service robot for home assistance applications," *Machines*, vol. 9, no. 3, p. 52, Mar. 2021.
[122] U. Rosolia, X. Zhang, and F. Borrelli, "Robust learning model-predictive control for linear systems performing iterative tasks," *IEEE Trans. Autom. Control*, vol. 67, no. 2, pp. 856–869, Feb. 2022.
[123] X. Qing, J. Song, J. Jin, and S. Zhao, "Nonlinear model predictive control for distributed parameter systems by time–space-coupled model reduction," *AIChE J.*, vol. 67, no. 8, p. e17246, Aug. 2021.
[124] M. N. Katehakis and A. F. Veinott, "The multi-armed bandit problem: Decomposition and computation," *Math. Operations Res.*, vol. 12, no. 2, pp. 262–268, May 1987.
[125] A. G. Barto, R. S. Sutton, and P. S. Brouwer, "Associative search network: A reinforcement learning associative memory," *Biol. Cybern.*, vol. 40, no. 3, pp. 201–211, May 1981.
[126] J. Berberich, J. Kohler, M. A. Müller, and F. Allgower, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021.
[127] H. Zhang, S. Seal, D. Wu, B. Boulet, F. Bouffard, and G. Joos, "Data-driven model predictive and reinforcement learning based control for building energy management: A survey," 2021, *arXiv:2106.14450*.
[128] F. Mahmood, R. Govindan, A. Bermak, D. Yang, C. Khadra, and T. Al-Ansari, "Energy utilization assessment of a semi-closed greenhouse using data-driven model predictive control," *J. Cleaner Prod.*, vol. 324, Nov. 2021, Art. no. 129172.
[129] K. Jiang, M. Kheradmandi, C. Hu, S. Pal, and F. Yan, "Data-driven fault tolerant predictive control for temperature regulation in data center with rack-based cooling architecture," *Mechatronics*, vol. 79, Nov. 2021, Art. no. 102633.
[130] S. Mirhoseininejad, G. Badawy, and D. G. Down, "A data-driven, multi-setpoint model predictive thermal control system for data centers," *J. Netw. Syst. Manage.*, vol. 29, no. 1, pp. 1–22, Jan. 2021.
[131] G. Torrente, E. Kaufmann, P. Fohn, and D. Scaramuzza, "Data-driven MPC for quadrotors," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3769–3776, Apr. 2021.
[132] A. Carron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, and M. N. Zeilinger, "Data-driven model predictive control for trajectory tracking with a robotic arm," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3758–3765, Oct. 2019.
[133] G. O. Guardabassi and S. M. Savaresi, "Virtual reference direct design method: An off-line approach to data-based control system design," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 954–959, May 2000.
[134] A. Karimi, K. van Heusden, and D. Bonvin, "Non-iterative data-driven controller tuning using the correlation approach," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2007, pp. 5189–5195.
[135] M. C. Campi and S. M. Savaresi, "Direct nonlinear control design: The virtual reference feedback tuning (VRFT) approach," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 14–27, Jan. 2006.
[136] M. Nakamoto, "An application of the virtual reference feedback tuning for an MIMO process," in *Proc. SICE Annu. Conf.*, vol. 3, Aug. 2004, pp. 2208–2213.
[137] A. Sala and A. Esparza, "Extensions to virtual reference feedback tuning: A direct method for the design of feedback controllers," *Automatica*, vol. 41, no. 8, pp. 1473–1476, Aug. 2005.
[138] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: Theory and applications," *IEEE Control Syst.*, vol. 18, no. 4, pp. 26–41, Aug. 1998.
[139] G. Venter, "Review of optimization techniques," in *Encyclopedia of Aerospace Engineering*. Hoboken, NJ, USA: Wiley, 2010, doi: 10.1002/9780470686652.eae495.
[140] J. Nocedal and Y.-X. Yuan, "Analysis of a self-scaling quasi-Newton method," *Math. Program.*, vol. 61, nos. 1–3, pp. 19–37, Aug. 1993.
[141] A. Karimi, L. Mišković, and D. Bonvin, "Iterative correlation-based controller tuning," *Int. J. Adapt. Control Signal Process.*, vol. 18, no. 8, pp. 645–664, Oct. 2004.

[142] L. Mišković, A. Karimi, D. Bonvin, and M. Gevers, "Correlation-based tuning of decoupling multivariable controllers," *Automatica*, vol. 43, no. 9, pp. 1481–1494, Sep. 2007.
[143] H. Hjalmarsson, "Iterative feedback tuning-an overview," *Int. J. Adapt. control signal Process.*, vol. 16, no. 5, pp. 373–395, 2002.
[144] S. Formentin, S. M. Savaresi, and L. del Re, "Non-iterative direct data-driven controller tuning for multivariable systems: Theory and application," *IET Control Theory Appl.*, vol. 6, no. 9, pp. 1250–1257, Jun. 2012.
[145] Z. Hou, C. Han, and W. Huang, "The model-free learning adaptive control of a class of MISO nonlinear discrete-time systems," *IFAC Proc. Volumes*, vol. 31, no. 25, pp. 227–232, Sep. 1998.
[146] A. S. Takialddin, O. I. Al-Agha, and K. A. Alsmadi, "Overview of model free adaptive (MFA) control technology," *IAES Int. J. Artif. Intell.*, vol. 7, no. 4, pp. 165–169, 2018.
[147] Z. Hou and W. Huang, "The model-free learning adaptive control of a class of SISO nonlinear systems," in *Proc. Amer. Control Conf.*, vol. 1, Jun. 1997, pp. 343–344.
[148] R. H. Chi and Z. S. Hou, "Dual-stage optimal iterative learning control for nonlinear non-affine discrete-time systems," *Acta Autom. Sinica*, vol. 33, no. 10, pp. 1061–1065, 2007.
[149] Z. Hou and S. Jin, "A novel data-driven control approach for a class of discrete-time nonlinear systems," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 6, pp. 1549–1558, Nov. 2011.
[150] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2173–2188, Dec. 2011.
[151] J. Descusse and C. H. Moog, "Decoupling with dynamic compensation for strong invertible affine non-linear systems," *Int. J. Control*, vol. 42, no. 6, pp. 1387–1398, Dec. 1985.
[152] S. Shastry and M. Bodson, "Adaptive control: Stability," in *Convergence and Robustness*. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.
[153] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*, vol. 291. New York, NY, USA: Springer, 2003.
[154] J.-X. Xu and Z.-S. Hou, "Learning control: The state of the art and perspective," Nat. Univ. Singapore Staff Publication Library, Singapore, 2005. [Online]. Available: https://scholarbank.nus.edu.sg/handle/10635/56475
[155] G. Honderd, "Iterative learning control for deterministic systems," *Automatica*, vol. 32, no. 6, pp. 948–949, Jun. 1996.
[156] C.-W. Chen and T.-C. Tsao, "Data-driven progressive and iterative learning control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4825–4830, 2017.
[157] R. Chi, Y. Wei, W. Yao, and J. Xing, "Observer-based data-driven iterative learning control," *Int. J. Syst. Sci.*, vol. 51, no. 13, pp. 2343–2359, Oct. 2020.
[158] F. Padieu and R. Su, "An H_∞ approach to learning control systems," *Int. J. Adapt. Control Signal Process.*, vol. 4, no. 6, pp. 465–474, 1990.
[159] G. Heinzinger, D. Fenwick, B. Paden, and F. Miyazaki, "Stability of learning control with disturbances and uncertain initial conditions," *IEEE Trans. Autom. Control*, vol. 37, no. 1, pp. 110–114, Jan. 1992.
[160] Y.-J. Liang and D. P. Looze, "Performance and robustness issues in iterative learning control," in *Proc. 32nd IEEE Conf. Decis. Control*, Dec. 1993, pp. 1990–1995.
[161] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proc.-Control Theory Appl.*, vol. 143, no. 2, pp. 217–224, 1996.
[162] S. Schaal and C. G. Atkeson, "Robot juggling: Implementation of memory-based learning," *IEEE Control Syst.*, vol. 14, no. 1, pp. 57–71, Feb. 1994.
[163] D. W. Aha, "The omnipresence of case-based reasoning in science and application," *Knowl.-Based Syst.*, vol. 11, nos. 5–6, pp. 261–273, Nov. 1998.
[164] R. S. Sutton, "Reinforcement learning: Past, present and future," in *Proc. Asia–Pacific Conf. Simulated Evol. Learn.* Florham Park, NJ, USA: Springer, 1998, pp. 195–197.
[165] S. G. Khan, G. Herrmann, F. L. Lewis, T. Pipe, and C. Melhuish, "Reinforcement learning and optimal adaptive control: An overview and implementation examples," *Annu. Rev. Control*, vol. 36, no. 1, pp. 42–59, Apr. 2012, doi: 10.1016/j.arcontrol.2012.03.004.
[166] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2005, pp. 319–324.
[167] R. S. Sutton, "Temporal credit assignment in reinforcement learning," Ph.D. dissertation, Univ. Massachusetts Amherst, Amherst, MA, USA, 1984, p. 223. [Online]. Available: https://www.proquest.com/dissertations-theses/temporal-credit-assignment-reinforcement-learning/docview/303321395/se-2?accountid=15083
[168] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. IEEE Amer. Control Conf.*, vol. 3, Jul. 1994, pp. 3475–3479.
[169] Y. Wang, F. Gao, and F. J. Doyle, "Survey on iterative learning control, repetitive control, and run-to-run control," *J. Process Control*, vol. 19, no. 10, pp. 1589–1600, Dec. 2009.
[170] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, FL, USA: CRC Press, 2017.
[171] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[172] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, Mar. 2007.
[173] A. Weissensteiner, "A *Q*-learning approach to derive optimal consumption and investment strategies," *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1234–1243, Jun. 2009.
[174] J.-H. Kim and F. L. Lewis, "Model-free H_∞ control design for unknown linear discrete-time systems via q-learning with lmi," *Automatica*, vol. 46, no. 8, pp. 1320–1326, 2010.
[175] R. Bellman, "A Markovian decision process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679–684, Apr. 1957.
[176] R. J. Boucherie and N. M. Van Dijk, *Markov Decision Processes in Practice*. Cham, Switzerland: Springer, 2017.
[177] G. Pannocchia, M. Gabiccini, and A. Artoni, "Offset-free MPC explained: Novelties, subtleties, and applications," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 342–351, 2015.
[178] C. Li and M. Qiu, *Reinforcement Learning for Cyber-Physical Systems: With Cybersecurity Case Studies*. New York, NY, USA: Chapman & Hall, 2019.
[179] N. Heess, G. Wayne, D. Silver, T. Lillicrap, Y. Tassa, and T. Erez, "Learning continuous control policies by stochastic value gradients," 2015, *arXiv:1510.09142*.
[180] M. Denil, P. Agrawal, T. D Kulkarni, T. Erez, P. Battaglia, and N. de Freitas, "Learning to perform physics experiments via deep reinforcement learning," 2016, *arXiv:1611.01843*.
[181] Z. Li, T. Wu, J. Na, J. Zhao, G. Gao, and G. Herrmann, "Data-driven based optimal output-feedback control of continuous-time systems," in *Proc. 10th Int. Conf. Model., Identificat. Control (ICMIC)*, 2018, pp. 1–6.
[182] C. J. C. H. Watkins, "Learning from delayed rewards," M.S. thesis, King's College, Cambridge, U.K., 1989.
[183] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[184] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.
[185] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[186] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.
[187] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[188] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, pp. 1–8, 2018.
[189] A. S. Takialddin, O. I. Al-Agha, and K. A. Alsmadi, "Overview of model free adaptive (MFA) control technology," *IAES Int. J. Artif. Intell.*, vol. 7, no. 4, p. 165, Oct. 2018.
[190] T. Martinetz and K. Schulten, "A neural network for robot control: Cooperation between neural units as a requirement for learning," *Comput. Electr. Eng.*, vol. 19, no. 4, pp. 315–332, Jul. 1993.

[191] S. M. Prabhu and D. P. Garg, "Artificial neural network based robot control: An overview," *J. Intell. Robotic Syst.*, vol. 15, no. 4, pp. 333–365, Apr. 1996.

[192] J. C. Spall and J. A. Cristion, "Model-free control of nonlinear stochastic systems with discrete-time measurements," *IEEE Trans. Autom. Control*, vol. 43, no. 9, pp. 1198–1210, Sep. 1998.

[193] K. B. Ariyur and M. Krstic, *Real-Time Optimization by Extremum-Seeking Control*. Hoboken, NJ, USA: Wiley, 2003.

[194] A. Scheinker and M. Krstić, *Model-Free Stabilization by Extremum Seeking*. Cham, Switzerland: Springer, 2017.

[195] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.

[196] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 657–667, Jan. 2015.

[197] S. Levine, "Exploring deep and recurrent architectures for optimal control," 2013, *arXiv:1311.1761*.

[198] Y. Lv, J. Na, Q. Yang, X. Wu, and Y. Guo, "Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics," *Int. J. Control*, vol. 89, no. 1, pp. 99–112, 2016.

[199] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[200] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 3, pp. 297–330, May 2020.

[201] O. H. Abu-Rub, A. Y. Fard, M. F. Umar, M. Hosseinzadehtaher, and M. B. Shadmands, "Towards intelligent power electronics-dominated grid via machine learning techniques," *IEEE Power Electron. Mag.*, vol. 8, no. 1, pp. 28–38, Mar. 2021.

[202] H. Hua, Y. Qin, C. Hao, and J. Cao, "Optimal energy management strategies for energy internet via deep reinforcement learning approach," *Appl. Energy*, vol. 239, pp. 598–609, Apr. 2019.

[203] K. Mason and S. Grijalva, "A review of reinforcement learning for autonomous building energy management," *Comput. Electr. Eng.*, vol. 78, pp. 300–312, Sep. 2019.

[204] M. Sarkar, A. Homaifar, B. A. Erol, M. Behniapoor, and E. Tunstel, "PIE: A tool for data-driven autonomous UAV flight testing," *J. Intell. Robotic Syst.*, vol. 98, no. 2, pp. 421–438, May 2020.

[205] K. Zhang, R. Su, H. Zhang, and Y. Tian, "Adaptive resilient event-triggered control design of autonomous vehicles with an iterative single critic learning framework," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5502–5511, Dec. 2021.

[206] Y. Li, W. Gao, W. Yan, S. Huang, R. Wang, V. Gevorgian, and D. Gao, "Data-driven optimal control strategy for virtual synchronous generator via deep reinforcement learning approach," *J. Modern Power Syst. Clean Energy*, vol. 9, no. 4, pp. 919–929, 2021.

[207] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, no. 7, pp. 1–53, 2009.

[208] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. Int. Conf. Artif. Neural Netw.* Springer, 2018, pp. 270–279.

[209] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 293–321, May 1992.

[210] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[211] R. A. C. Bianchi, C. H. C. Ribeiro, and A. H. R. Costa, "Accelerating autonomous learning by using heuristic selection of actions," *J. Heuristics*, vol. 14, no. 2, pp. 135–168, Apr. 2008.

[212] M. F. Martins and R. A. Bianchi, "Heuristically-accelerated reinforcement learning: A comparative analysis of performance," in *Proc. Conf. Towards Auto. Robotic Syst.* Springer, 2013, pp. 15–27.

[213] S. Hochreiter, A. S. Younger, and P. R. Conwell, "Learning to learn using gradient descent," in *Artificial Neural Networks—ICANN 2001*, G. Dorffner, H. Bischof, and K. Hornik, Eds. Berlin, Germany: Springer, 2001, pp. 87–94.

[214] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "RL^2: Fast reinforcement learning via slow reinforcement learning," 2016, *arXiv:1611.02779*.

[215] Y. Wang, V. Kirubakaran, and H. Biao, "A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems," *Processes*, vol. 5, no. 3, p. 46, Sep. 2017.

[216] R. Goel and S. B. Roy, "Closed-loop reference model based distributed model reference adaptive control for multi-agent systems," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 1082–1087.

[217] *Openai*. *Openai/Gym*. Accessed: Jan. 25, 2022. [Online]. Available: https://github.com/openai/gym/wiki/CartPole-v0

[218] P. Christiano, Z. Shah, I. Mordatch, J. Schneider, T. Blackwell, J. Tobin, P. Abbeel, and W. Zaremba, "Transfer from simulation to real world through learning deep inverse dynamics model," 2016, *arXiv:1610.03518*.

[219] M. Cutler, T. J. Walsh, and J. P. How, "Reinforcement learning with multi-fidelity simulators," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3888–3895.

[220] M. Cutler and J. P. How, "Efficient reinforcement learning for robots using informative simulated priors," *Proc. IEEE Int. Conf. Robot. Autom.*, Jun. 2015, pp. 2605–2612.

[221] M. Cutler, T. J. Walsh, and J. P. How, "Real-world reinforcement learning via multifidelity simulators," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 655–671, Jun. 2015.

[222] A. Plasencia, Y. Shichkina, I. Suárez, and Z. Ruiz, "Open source robotic simulators platforms for teaching deep reinforcement learning algorithms," *Proc. Comput. Sci.*, vol. 150, pp. 162–170, Jan. 2019.

[223] I. V. Serban, C. Sankar, M. Pieper, J. Pineau, and Y. Bengio, "The bottleneck simulator: A model-based deep reinforcement learning approach," *J. Artif. Intell. Res.*, vol. 69, pp. 571–612, Oct. 2020.

[224] S. Haag and R. Anderl, "Digital twin – proof of concept," *Manuf. Lett.*, vol. 15, pp. 64–66, Jan. 2018, doi: 10.1016/j.mfglet.2018.02.006.

[225] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[226] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020.

[227] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *J. Manuf. Syst.*, vol. 48, pp. 157–169, Jul. 2018, doi: 10.1016/j.jmsy.2018.01.006.

[228] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, "Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison," *Engineering*, vol. 5, no. 4, pp. 653–661, Aug. 2019, doi: 10.1016/j.eng.2019.01.014.

[229] D. Ivanov and A. Dolgui, "A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0," *Prod. Planning Control*, vol. 32, pp. 775–788, May 2020, doi: 10.1080/09537287.2020.1768450.

[230] K. Y. H. Lim, P. Zheng, and C.-H. Chen, "A state-of-the-art survey of digital twin: Techniques, engineering product lifecycle management and business innovation perspectives," *J. Intell. Manuf.*, vol. 31, pp. 1313–1337, Nov. 2019, doi: 10.1007/s10845-019-01512-w.

[231] A. Verma, V. Murali, R. Singh, P. Kohli, and S. Chaudhuri, "Programmatically interpretable reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 11, 2018, pp. 8024–8033.

[232] R. Noothigattu, D. Bouneffouf, N. Mattei, R. Chandra, P. Madan, K. Varshney, M. Campbell, M. Singh, and F. Rossi, "Interpretable multi-objective reinforcement learning through policy orchestration," 2018, *arXiv:1809.08343*.

[233] E. Puiutta and E. M. S. P. Veith, "Explainable reinforcement learning: A survey," in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds. Cham, Switzerland: Springer, 2020, pp. 77–95.

[234] P. Linardatos, V. S. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, pp. 1–45, 2021.

[235] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou, "Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond," 2021, *arXiv:2103.10689*.

[236] Z. Hou and S. Xiong, "On model-free adaptive control and its stability analysis," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4555–4569, Nov. 2019.

[237] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," 2020, *arXiv:2009.07888*.
[238] C. Vallon and F. Borrelli, "Data-driven hierarchical predictive learning in unknown environments," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2020, pp. 104–109.
[239] S. Boschert and R. Rosen, "Digital twin—The simulation aspect," in *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers*, P. Hehenberger and D. Bradley, Eds. Cham, Switzerland: Springer, 2016, pp. 59–74, doi: 10.1007/978-3-319-32156-1_5.

KRUPA PRAG is currently a Postgraduate Student at the University of the Witwatersrand, Johannesburg, South Africa, where she is also an Associate Lecturer with the School of Computer Science and Applied Mathematics. Her research interests include optimization, optimal control theory, and computational intelligence.

MATTHEW WOOLWAY received the Ph.D. degree in process engineering from the University of the Witwatersrand, Johannesburg, South Africa. He is currently an Industry Data Scientist and a Research Associate with the Faculty of Engineering and the Built Environment, University of Johannesburg. His research interests include computational intelligence, artificial intelligence, and optimization.

TURGAY CELIK received the Ph.D. degree from the University of Warwick, Coventry, U.K., in 2011. He is currently a Professor of digital transformation and the Director of the Wits Institute of Data Science, University of the Witwatersrand, Johannesburg, South Africa. His research interests include signal and image processing, computer vision, machine intelligence, robotics, data science, and remote sensing. He is an Associate Editor of *ELL* (IET), IEEE Access, IEEE Geoscience and Remote Sensing Letters, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, and *SIVP* (Springer).