Direct Data-Driven Predictive Control

Francisco Fonseca, Sahil Salotra and Peng Xie

***Abstract*— In recent years data-driven methods have risen in popularity as a way to bypass model identification of complex robotic systems. Furthermore, robotic systems are becoming more complex and harder to model. Researchers have been exploring using data to bypass the model identification step and achieve robust control using Direct Data-Driven Predictive Control (DDDPC). The main approaches explored are linearization, learning-based methods, and persistent excitation. This survey aims to give a self-contained review of the theory behind DDDPC and an overview of the current approaches and implementations.**

I. Introduction

A. Motivation

The field of robotics has seen an increase in the complexity of the systems it aims to control, and the difficulty of modeling and controlling highly complex nonlinear robotic systems is evident [1]. Furthermore, expert knowledge is usually needed to study the system and get an accurate model for the system's dynamics [2]. In other cases, achieving an accurate model of the system is infeasible [2].

In recent years, researchers have explored the possibility of using data-driven methods to derive control methods that ensure robust and safe control of these robots to bypass the system identification step. Figure 1 shows the outline of the data-driven approach.

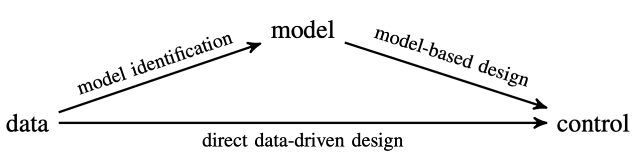

Fig. 1. The data-driven design paradigm aims to achieve control design from data either with the model identification step (indirect) or bypassing the model identification step (direct) [3].

The top line describes the indirect method, where the data is used for model identification. For more details about the indirect method, we refer to [4] and [5]. This approach has similar issues to the conventional model-based methods, such as high model complexity.
Another direction that could be considered in this data-driven design from Figure 1 is Reinforcement Learning (RL). RL deals with an agent interacting with an unknown environment to achieve his objective by maximizing the cumulative reward, which could be achieved by either a model-free- or model-based method [6].

Therefore, we will not do in-depth research about RL and its characteristics to label the methods as model-free or model-based but instead, focus on collecting all methods that fit the main topic in our survey. To be precise, this survey will focus on direct data-driven methods that bypass any model identification or dynamics description. Specifically, we review how DDDPC can be implemented for diverse robotics applications and discuss its advantages and disadvantages.

B. Problem Statement

We formulate the DDDPC problem as a classical optimal control problem with some key differences. The goal is to successfully control robotic systems with a guarantee of robustness (achieve desired performance in the presence of uncertainty) and safety (achieve goals while meeting safety constraints) that fulfills the required tasks. However, contrary to the model-based formulation, where the model of the system's dynamics is identified or known, the DDDPC, based on behavioral system theory, only uses the data to achieve the final goal of a reliable controller, which leads to data-related issues, mainly related to quality and availability [7]. This paper will explore how the literature has approached this problem and how the different solutions compare.

C. Structure

The paper is structured as follows. In Section II and Section III, we present background theory and explain the fundamental concepts of data-driven control (DDC). In Section IV, we explore different DDDPC approaches in the literature, focusing on robotic systems. In the last part, Section V discusses the result of the latest DDDPC approaches in simulation and hardware and concludes where the field currently stands.

II. Background

A. Behavioral System Theory

Behavioral System Theory is an approach [7] that focuses on analyzing and understanding the behavior of dynamic systems without relying on explicit mathematical models. It emphasizes empirical data, observations, and measurements to describe and analyze the system's behavior. Instead of traditional mathematical models, behavioral models are used, which describe the system's input-output relationships based on experimental data.

One of the critical features of Behavioral System Theory is its "black-box" perspective, treating the system as an entity with inputs and outputs without requiring detailed knowledge of the internal mechanisms or physical processes, which allows for a non-parametric description of the system [8], enabling the analysis of complex and nonlinear systems without the need for extensive mathematical derivations.

Behavioral System Theory is particularly useful for studying dynamic behavior, capturing time-varying responses and transient and steady-state behavior [7]. It often employs data-driven methods, such as system identification techniques from signal processing and machine learning, to estimate the input-output relationships based on empirical data. Behavioral system theory has a strong connection with data-driven control. Here we say data-driven control is split into two types: direct and indirect data-driven control. The latter can also be treated as model-based control under some conditions.

According to [7], the field of Behavioral System Theory, once considered an unconventional and specialized research niche, has experienced a resurgence of interest in light of the emerging data-driven paradigm. This renewed attention stems from the unique suitability of Behavioral System Theory for the data-driven approach, thanks to its representation-free perspective and the availability of advanced computational methods.

A significant breakthrough in the behavioral framework, known as the fundamental lemma [9], has played a pivotal role in paving the way for a new class of data-driven methods based on subspace techniques. The fundamental lemma provides conditions for a non-parametric representation of a Linear Time-Invariant (LTI) system using the image of a Hankel matrix constructed from raw time series data [10]. This robust result has opened up possibilities for leveraging data directly and without explicit models, enabling more flexible and adaptive approaches to system analysis, signal processing, and control.

B. Model Predictive Control

In recent decades, Model Predictive Control (MPC) has emerged as a highly successful and widely adopted control method across various fields, including process control, automotive systems, and robotics. Its effectiveness lies in systematically handling system constraints. To achieve this, MPC relies on a comprehensive model of the system, enabling optimization of performance while ensuring constraint satisfaction. However, uncertainty in model descriptions can arise in practical applications due to limited data, constrained model classes, or external disturbances. In [11], MPC has several significant features. MPC relies on a mathematical model of the system being controlled. The control strategy is based on predicting the system's behavior using this model. The quality of MPC's control performance heavily depends on the accuracy of the underlying model. A precise model is crucial for achieving optimal control. By incorporating them into the optimization process, MPC can handle control constraints, such as physical limitations or safety requirements. MPC inherently provides stability guarantees due to the optimization-based approach. However, robustness depends on the model's accuracy and ability to capture system variations. In [12], the model parameters often require system identification before implementing MPC, which may involve gathering experimental data.

C. Data-driven Control

The performance of DDC heavily depends on the quality and quantity of the data used for training the control algorithm [8]. Sufficient and diverse data are essential for effective learning [12]. DDC can handle linearity and weak nonlinearity in the system [8]. Sufficient and representative data are required to train the data-driven algorithms initially. Data collection and preparation are critical steps in the implementation. The stability and safety of DDC policies should be carefully validated, as they might not inherently guarantee stability like some model-based methods [10]. Usually, DDC includes two types, direct and indirect data-driven control. The distinction between direct and indirect methods can be related to whether the model is used [12]. Methods using parametric models, such as rational transfer function and state-space representation, are indirect, while non-parametric models, such as the data-driven representation presented in the paper, are direct.

D. Mathematical Foundations

Some significant notions will be widely used in the following article and are introduced in this section.

In [13], the LTI system considered is shown below,

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) \tag{1a}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t), \tag{1b}$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes the state, $\mathbf{u} \in \mathbb{R}^m$ is the input and $\mathbf{y} \in \mathbb{R}^p$ is the output. Let $\left(u_{[0,T-1]}, y_{[0,T-1]}\right)$ be a given input/output trajectory.

1) Hankel matrix: In [9], a clear definition of the Hankel matrix is provided. We directly cite the definition here for better reference. Consider a signal $f : \mathbb{Z} \to \mathbb{R}^\bullet$ and let $i, j \in \mathbb{Z}$ be integers such that $i \leq j$. We denote by $f_{[i,j]}$ the restriction of f to the interval $[i, j]$, that is,

$$f_{[i,j]} := \begin{bmatrix} f(i)^\top & f(i+1)^\top & \cdots & f(j)^\top \end{bmatrix}^\top .$$

With slight abuse of notation, we will also use the notation $f_{[i,j]}$ to refer to the sequence $f(i), f(i+1), \ldots, f(j)$. Let k be a positive integer such that $k \leq j - i + 1$ and define the Hankel matrix of depth k, associated with $f_{[i,j]}$, as

$$\mathcal{H}_k\left(f_{[i,j]}\right) := \begin{bmatrix} f(i) & f(i+1) & \cdots & f(j-k+1) \\ f(i+1) & f(i+2) & \cdots & f(j-k+2) \\ \vdots & \vdots & & \vdots \\ f(i+k-1) & f(i+k) & \cdots & f(j) \end{bmatrix}.$$

Note that the subscript k refers to the number of block rows of the Hankel matrix.

2) persistent excitation: In [14], some notions are based on persistent excitation in DC, which plays a vital role in data-driven control. In order to complement the data-driven control, persistent excitation must be satisfied first. Furthermore, how to filter input data that satisfies such conditions will also be mentioned in Section IV. We say the matrix has the rank condition:

$$\text{rank}\begin{bmatrix} \dfrac{U_{0,t,T-t+1}}{[X_{0,T-t+1}]} \end{bmatrix} = n + tm$$

As we delve further, we will discover that a condition of this nature guarantees that the data encapsulates all the necessary information for the direct design of control laws. The essential aspect is ensuring the rank when the input is adequately stimulating. Before proceeding, let us review the concept of persistent excitation.

The signal $z_{[0,T-1]} \in \mathbb{R}^{\sigma}$ is persistently exciting of order L if the matrix

$$Z_{0,L,T-L+1} = \begin{bmatrix} z(0) & z(1) & \cdots & z(T-L) \\ z(1) & z(2) & \cdots & z(T-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ z(L-1) & z(L) & \cdots & z(T-1) \end{bmatrix}$$

has full rank σL.

3) Willems' Fundamental Lemma: Consider a linear time-invariant system $B \in L^q$ with an input/output partition $w = (u, y)$ [13].

Let $w_d = (u_d, y_d) \in B|_T$ be a trajectory of B, and the system B be controllable, and the input component u_d of w_d be persistently exciting of order $L + \mathbf{n}(B)$. Then any t-long input/output trajectory of the system can be expressed as

$$\begin{bmatrix} u_{[0,t-1]} \\ y_{[0,t-1]} \end{bmatrix} = \begin{bmatrix} U_{0,t,T-t+1} \\ \hline Y_{0,t,T-t+1} \end{bmatrix} g$$

where $g \in \mathbb{R}^{T-t+1}$. Any linear combination of the columns of the matrix that is

$$\begin{bmatrix} U_{0,t,T-t+1} \\ \hline [Y_{0,t,T-t+1}] \end{bmatrix} g$$

is a t-long input/output trajectory.

III. Direct Data-driven Control

To obtain a clearer and more intuitive appreciation of the physical significance of direct data-driven control, a new model has been proposed by C. De Persis and colleagues at the University of Zurich. This model facilitates enhanced research and simulation of the method in question. Furthermore, the following figure 2 can be used to illustrate the theorem.

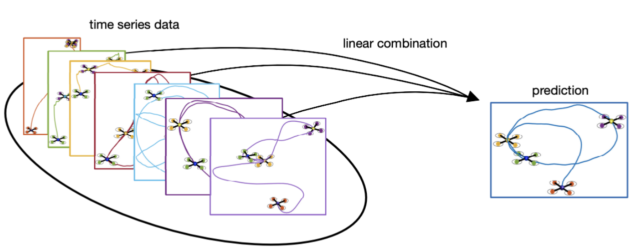

Fig. 2. The function of linear combination [3]

The theory put forth in [3] suggests using prior trajectories to estimate future trajectories. The linear combination serves as a weighting factor, representing the effect of each trajectory at various times on the future trajectory. We use the Hankel matrix to represent the previous trajectories.

Let now $u_{d,[0,T-1]}$ and $y_{d,[0,T-1]}$ be the input-output data of the system collected during an experiment and let

$$[U : Y] := \begin{bmatrix} u_d(0) & u_d(1) & \cdots & u_d(T-t) \\ u_d(1) & u_d(2) & \cdots & u_d(T-t+1) \\ \vdots & \vdots & \ddots & \vdots \\ u_d(t-1) & u_d(t) & \cdots & u_d(T-1) \\ \hline y_d(0) & y_d(1) & \cdots & y_d(T-t) \\ y_d(1) & y_d(2) & \cdots & y_d(T-t+1) \\ \vdots & \vdots & \ddots & \vdots \\ y_d(t-1) & y_d(t) & \cdots & y_d(T-1) \end{bmatrix}$$

be the corresponding Hankel matrix, here, $U = U_{0,t,T-t+1}$, $Y = Y_{0,t,T-t+1}$.

Then any t-long input/output trajectory of the system can be expressed as

$$\begin{bmatrix} u_{[0,t-1]} \\ y_{[0,t-1]} \end{bmatrix} = \begin{bmatrix} U_{0,t,T-t+1} \\ \hline Y_{0,t,T-t+1} \end{bmatrix} g$$

where $g \in \mathbb{R}^{T-t+1}$ [3].

Each column represents a trajectory; we use the initial state to calculate the linear combination g and predict the next step. The algorithm will be:

Given the initial state $y_{[0,0]}, u_{[0,N]}, \bar{u}_{[0,N]}, \bar{y}_{[0,N]}$

- Step 1: given $y_{[0,0]}, u_{[0,T-1]}$ and use it to generate the $\mathcal{H}_L(u_{[0,T-1]}, y_{[0,0]})$
- Step 2: calculate the g via the pseudoinverse of $\mathcal{H}_L(u_{[0,T-1]}, y_{[0,0]})$
- Step 3: calculate the $y_{[1,T]}$, and put $y_{[T,T]}$ as the initial value to replace $y_{[0,0]}$, back to the step1, till get all $y_{[0,N]}$

A. Data-Driven Control Design

In order to be able to compare the DDDC method with other methods, we first need to specify a criterion, i.e., how large the error is between the result obtained using the DDDC method and the real result.Here, we still focus on the LTI system (1a) and (1b)

In direct data-driven control design, we would like to use the input-output behavior data matrix $\begin{bmatrix} U^\top & Y^\top \end{bmatrix}^\top$ directly in place of the model. The key idea here is to avoid identifying a model of input-output behaviors and to directly search for the optimal behavior within the span of observed behaviors contained in the data matrix $\begin{bmatrix} U^\top & Y^\top \end{bmatrix}^\top$.

In [12], it establishes that given the input-output model $\mathcal{G}$, the minimizer to the stochastic optimization problem can be equivalently obtained as the solution to the following deterministic optimization problem:

$$\min_{\mathbf{u} \in \mathbb{R}^{mT}} F(\mathcal{P}\mathbf{u}), \quad \mathcal{P} = \begin{bmatrix} I \\ \mathcal{G} \end{bmatrix}.$$

where

$$\mathcal{G} = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{T-1}B & CA^{T-2}B & \ldots & CB \end{bmatrix}.$$

The main idea is to compute the minimizer $\mathbf{u}$.

With certainty equivalent model-based predictive control [12], we can find the solution of u. Moreover, after we add the noise into the data, we can find the optimal solution to $\widehat{\mathbf{u}}_{\text{direct}}$. We note that the control input $\widehat{\mathbf{u}}$ is not guaranteed to be optimal for the control task, owing to the mismatch between the estimate $\widehat{\mathcal{G}}$ and the true model $\mathcal{G}$. In [12], the performance of $\widehat{\mathbf{u}}$ is measured by its suboptimality gap given by:

$$\operatorname{Gap}(\widehat{\mathbf{u}}) = F(\mathcal{P}\widehat{\mathbf{u}}) - F(\mathcal{P}\mathbf{u}).$$

$$\Delta_{\text{direct}} = \widehat{\mathcal{G}}_{\text{direct}} - \mathcal{G}$$

So here we can calculate $\operatorname{Gap}(\widehat{\mathbf{u}})$ to compare the data-driven control method with other methods.
After conducting a mathematical analysis of the two methods based on the previously stated theory, an experimental method has been put forward to evaluate the superiority of two distinct techniques.

B. The Process of the Experiment

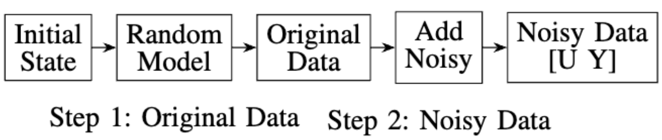

Fig. 3. The flow of the experiment design

According to the theory mentioned in [12], if we want to achieve such a goal, we think about We proceed by utilizing an available model to generate data, which is subsequently subjected to the introduction of noise. Subsequently, we apply two methods to the resultant noisy data and compare the outcomes with the original data. The criterion for judgment is the error gap $\operatorname{Gap}(\widehat{\mathbf{u}})$ mentioned above.
According to the error gap, we can determine which method is better.

C. The factors influencing Direct Data-driven Control's performance

Also, in [12], several essential factors mentioned will influence the performance of DDDC, including the Datasize N, length of the trajectory T, and noise intensity.

Furthermore, [12] provides equations to describe the results. The implicit model error $\Delta_{\text{direct, given}}$, satisfies:

$$\mathbb{E}\left[\Delta_{\text{direct}}\right] = 0,$$

$$\mathbb{P}\left\{\left\|\Delta_{\text{direct}}\right\|_F \geqslant \epsilon\right\} \leqslant \frac{T^2}{N\epsilon^2} \cdot C,$$

where C is the function of smallest singular value of Σ_{uu}^N and the variance of input u.

According to Figures 4 and 5 [12], we can get the following conclusion.

The dataset size is bigger, the performance of DDDC is better, and if the dataset is big enough the error trends to 0. Regarding the trajectory length T indirect method, the result

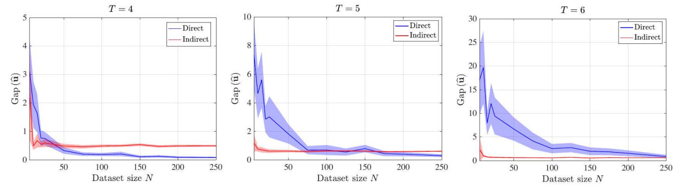

Fig. 4. The influence of trajectory length T [12]

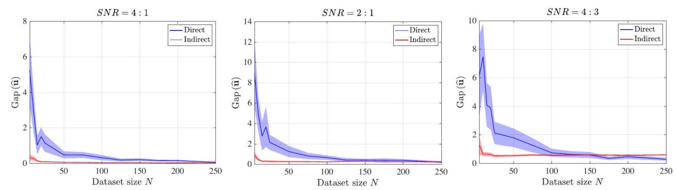

Fig. 5. The influence of signal-to-noise SNR [12]

is that T is small. The result is more accurate, but more steps are needed, and the process is slower. The process is faster if T is big, but we need a more extensive data size for the convergence. When it comes to the Signal-to-Noise SNR, the result can be concluded that the Direct method performs better and $\mathbf{error}_{\text{direct}}$ tends to 0 with data size increasing. Moreover, DDDC's performance gets less impact when noise intensity increases than the indirect method.

IV. Direct Data-driven Predictive Control Approaches

A. Linearization

Linearization is a common and popular method used to deal with nonlinear control systems using mathematical tools to simplify the design of nonlinear complex systems [15]. This section will explore the different approaches used for linearization in the context of DDDPC.

1) Feedback Linearization: Feedback linearization is a common approach for controlling nonlinear systems. It involves transforming the nonlinear system into an equivalent linear system through state feedback, which can then be controlled using linear control techniques.

In [16], a data-driven model-free direct adaptive control (DDMFDANPC) approach is applied to linear motor position control. This approach leverages a generalized predictive control (GPC) for linearizing a class of Singular Input Singular Output (SISO) nonlinear systems. The controller design is based directly on the online estimation and prediction of pseudo-partial derivatives (PPD) derived from the input and output information of the motor motion model, which is achieved using a novel parameter estimation algorithm and a multi-degree prediction approach. The authors demonstrated the system's stability, validity, and robustness against disturbances through experimental research. Similarly, in [17], an algorithm was developed to control a class of SISO nonlinear systems. The linearization approach allows us to compute a robust control invariant (RCI) set to obtain the associated state-feedback control laws. It was shown that the designed algorithm is robust to all models compatible with the available data.

Another approach for nonlinear systems is proposed in [18], which presents a method for tuning the parameters of a look-up table (LUT) in gain-scheduled feedback controllers. They introduced a direct data-driven design method that enables the tuning of LUT parameters from single-experiment data, eliminating the need for a system model. This method extends the conventional virtual reference feedback tuning (VRFT), a data-driven method, and introduces the L_2 norm for adjacent LUT parameters to the VRFT cost function to prevent overlearning. The optimized parameters are obtained analytically through a generalized ridge regression. This approach was extended to two-degree-of-freedom control in [19], with the ability to realize a desired closed-loop response of a non-minimum phase system.

In [20], the fundamental lemma is used to design a direct data-driven state-feedback control method for a general class of nonlinear systems, which is achieved by exploiting a linear parameter-varying (LPV) framework extension to the fundamental lemma. It achieves universal stability and dissipativity. Similarly, in [21], the linearization is made with a linear time-varying (LTV) state feedback controller to achieve locomotion control for a snake-robot.

A common thread across papers that focused on feedback linearization is that for systems with high complexity and hard-to-model dynamics, the direct data-driven methods showed significant performance improvements, as we will see in the results of Section V.

2) Koopman Operators: The Koopman operator has emerged as a powerful tool for analyzing and controlling nonlinear dynamical systems. It provides a linear representation of the dynamics in a high-dimensional function space, which can benefit the design and analysis of control systems.

In [22], a direct data-driven controller for nonlinear affine systems using the Koopman operator is proposed. They showed how the Koopman operator could be used to reformulate nonlinear systems into bilinear forms and how this can be used to design a simple static controller.

The robustness of systems using the Koopman operator has also been explored. In [23], a method to quantify the prediction error due to noisy measurements is proposed. The Koopman operator is approximated via Extended Dynamic Mode Decomposition and is used to develop a control strategy for robotic data-driven systems. It is concluded through tests with a simulated non-holonomic wheeled robot that to have a real-time implementation, data has to be captured offline.

These papers show the advantages of the Koopman operator for simplifying the control design process for nonlinear systems.

B. Learning-based Methods

Another way to replace the traditional model description of a system while still achieving control is Neural Networks (NNs) [24]. NNs focus on mimicking the system's behavior [25] rather than generating an exact system description, which falls under the DDDPC domain. It is important to note that the setup of the NNs is crucial to label a method as DDDPC. For example, in [24], a multi-layer neural network was trained on an I/O dataset from an aerial robot controlled by PID, but since the weights of the NN do not update during the control process by the data being generated, this approach did not use the data for the control directly and from our point of view and can not be seen as a DDDPC method. Also, in [26], the DeLAN (Deep Lagrangian Network) [27] is used to learn the underlying ODE of the system from the data, but since after pre-training the core of this NN is frozen, indicating it is not updating its weights, the data generated will not affect the control of the system. In the following, we will first discuss how motion control of a robot with NNs as DDDPC can be achieved. Here, we consider only the motion control of a robot and then motion control with additional robot manipulation. Second, a walking assistance control based on RL and a NN will be introduced.

1) Motion Control without Robot-Manipulation: First, a discrete zeroing neural network (DZNN) [28] used for data-driven motion control will be discussed. DZNNs are based on the well-known Recurrent Neural Networks (RNNs), which are not usable for time-variant problems due to their design. Therefore, a special type of RNN, the zeroing neural network (ZNN), was designed, which can be in continuous-time (CZNN) or discrete-time (DZNN), but the latter is more suitable for hardware integration. ZNNs generally utilize differential- or state-transition equations and consist of constant parameters; therefore, training with a dataset is unnecessary. [28] The control of the robot is achieved via two CZNNs online with real-time data from sensors. The first CZNN is used to force the error between the desired path and the actual position of the end-effector of the robot to converge to zero. Here, the CZNN is used to create an actuation signal. The second CZNN has a similar objective by forcing the error between the velocity of the end-effector and its estimated velocity to converge to zero. The proposed data-driven CZNN-based control algorithm only needs feedback on the end-effector's position, which is achieved with a low-pass filter to approximate the velocity and acceleration of the end-effector. [28]

Here, a NN can control a robot's motion purely on data.

2) Motion Control with Robot-Manipulation: A data-driven motion control via NNs was already discussed. In [29], the robot manipulation task via NN involves additionally grasping the object. It is important to note here that in contrast to the data-driven control approach with a NN mentioned before, in [29], a dataset of suitable action is generated by observation from human behavior. To be concise, from videos of humans grasping a different object, a dataset of reactive primitives to mimic human behavior is generated. The NN, adapted and based on the Inception-v3 [30], predicts a suitable action from the dataset containing the human reactive primitives based on the object it should grasp. Before predicting one of the primitives of how humans would grasp the object, object detection and classification is done by the YOLOv2 [31] detector. The method for data-driven motion control with robot manipulation based on NN was shown, where in [29], a dataset was generated based on observation from humans with videos.

3) Walking Assistance Control: A Lower limb exoskeleton (LLE) must be controlled to achieve walking assistance for a hemiplegic patient. Here, the task is that the affected leg tries to track the motion of the unaffected one. One crucial point is that the controller needs to adapt to the different hemiplegic patients influenced by different individual disturbances. [32] To tackle this issue in [32] data-driven Reinforcement Learning (DDRL) method to control an LLE for walking assistance of hemiplegic patients was introduced. First, the DDRL learns the optimal walking assistance controller relying on a policy iteration (PI) algorithm, which is one of the fundamental methods used in RL. The PI algorithm needs the system model, which is not suitable for this approach, and therefore the system identification step needs to be fixed by the control strategy. As already mentioned, the aim is that the controller is robust to changes, which are either to control different hemiplegic patients with the same controller or disturbances. Also, every hemiplegic patient is faced with different disturbances. For this, the walking assistance controller adapts in an online-learning manner based on an ACNN (Actor-Critic Neural Network), which consists of the actor-network and critic network. The actor-network approximates the controller, while the critic network estimates the online parameters of the PI algorithm to find the optimal walking assistance controller. [32]. Since the weights of the ACNN get updated during the control, this is genuinely a DDDPC approach.

C. Dealing with persistent excitation

The emphasis on persistent excitation is justified because only persistently exciting data allow for constructing data-dependent matrices that can effectively substitute system models. [14] And persistently exciting data could be used to represent the input-output trajectory of a linear system. [33]

A direct implication of the fundamental lemma is that a persistently exciting trajectory captures the complete behavior of the data-generating system. This property enables the successful identification of a system model using subspace methods [34]. Persistency of excitation plays a crucial role in various aspects, including system identification and input design problems. [8]

Let us revisit the fundamental lemma, which states that if the input is persistently exciting of order $n+1$, then the input-state data matrix will have a full row rank [8]. This rank condition guarantees that the pseudo-inverse of the input-state data matrix serves as a right inverse. In practical terms, the measurement data equation can be solved to obtain the system parameters (B, A). [8].

In [35], it states that for any controllable Linear Time-Invariant (LTI) system, a set of discrete impulses is guaranteed to excite the system persistently. Consequently, designing the physical input alone can ensure a priori persistence of excitation for sequences of basis functions in certain classes of nonlinear systems.

However, ensuring persistent excitation in closed-loop scenarios, particularly during setpoint tracking, poses challenges as the continuous update of the data matrix cannot be guaranteed. [36] address this issue by introducing suitable excitation to the controller input during closed-loop operation, such as injecting noise. However, this approach can degrade control performance and unnecessarily strain actuators [8].

In order to solve such a problem, [8] introduces a novel online adaptation strategy for data-driven system representation, eliminating the need to impose specific inputs on the system during control operation. This method allows for pausing data updates when a setpoint is reached and resuming updates once the system is in motion again. It achieves this by using Hankel matrices [33] and an extended version of the fundamental lemma [37], enabling the utilization of discontinuous input-output trajectories for data-driven system representation.

The algorithm presented in [8] only requires a single initially measured persistently exciting input-output trajectory and updates the data-driven system representation at each time step based on an extended rank condition applied to the collected data matrix. To evaluate the rank condition in the presence of noise and/or nonlinearity in the data, the Singular Value Decomposition (SVD) of the data matrices is employed. Furthermore, this method can be independently applied to the data-driven design of state-feedback controllers [14]

V. Implementations and Results

This section will briefly overview the implementations and results of the approaches mentioned in IV.

A. DDDPC-Approach: Linearization

In Section IV-A, we explored the main approaches in the literature to apply DDDPC in robotics. We will now go through how researchers applied the theory and advancements proposed both as simulations and real implementations.

In [16], a practical implementation is shown. Direct data-driven methods control a linear motor based on cSPACE technology. The algorithm developed allows for real-time effective control and modification of the controller parameters. Furthermore, results show high robustness to disturbance and white noise compared to similar model-based controllers.

RCI sets are explored in [17] through a numerical analysis simulation of a double integrator system to compare the RCI sets' sizes from data-driven approaches and model-based methods. It was concluded that the RCI is more extensive for the direct data-driven method presented in the paper.

In [18], a simulation is made for a Hammerstein model. It is concluded that the use of the L2 norm was a success regarding the prevention of overlearning. It is noted, however, how no noise was assumed and might lead to the unfeasibility of the proposed methods. For non-minimum phase systems, in [19], it was shown that it is possible to achieve good control performance with DDDPC without updating the feedback and feedforward controllers.

To compare with LTI controllers, in [20], it is shown that an LTI controller diverges when controlling a simulation of an unbalanced disk, where the developed state-feedback controller for nonlinear systems can control with success with only one basis set assumed.

In [21], a simulation of a snake robot is developed. The algorithm developed in the paper showed better performance than model-based approaches due to the high complexity of the nonlinear system.

The conclusion that disturbance and noise cannot be too high for data-driven methods is also reached in [22], where numerical experiments show these results when controlling systems with polynomial dynamics. An inverted pendulum is also stabilized with an insufficient polynomial dictionary to show robustness.

In [23], two simulations explore the Koopman operator approach. The first simulation is of a Van der Pol Oscillator where the prediction error is tested for different data sizes, concluding that there is an optimal bound for this size. The second simulation is done in a Gazebo of a non-holonomic wheeled robot, where both a nominal and noisy controller are tested. The prediction error gets worse for big noises, but for small noises, it achieves the task of tracking the given trajectory.

B. DDDPC-Approach : Learning-Based

1) Motion Control without Robot-Manipulation: In Section IV-B.1, the method to apply purely data-driven motion control via NN was shown, and here we discuss their results. In [28], simulations and experiments were performed. In [28], two CZNNs were used to control a continuum robot to track a circular path. A generic forward m-step discretization formula was introduced to derive the 1-step or 3-step DZNN-based control method. The step number should be chosen according to the requirement for accuracy. For comparison, four model-free tracking algorithms, e.g., gradient-neural-network (GNN) [38] and continuous-time zeroing neuro-dynamic (CTZND) [39] were implemented to control the continuum robot and the root-mean-squared error (RMSE) metric used to evaluate their tracking error. In the simulation, the RSME of the four model-free tracking algorithms lies in the range between $1.2 \times 10^{-4}m$ and $8.3 \times 10^{-4}m$, while the 3-step DZNN-based control method has a much lower tracking error around $2.9 \times 10^{-7}m$. The 1-step DZNN-based control method was compared to a traditional model-based method, where the kinematic model from [40] was used. If the influence of model uncertainties of the model-based approach is considered, then the 1-step DZNN-based shows a lower tracking error since it is based on a model-free approach and, therefore, unaffected by uncertainties in the model description. [28]

The experiments were conducted on two different continuum robots. Again, the 3-step DZNN-based algorithm showed the lowest tracking error for a circular path with an RMSE around $1.2 \times 10^{-3}m$ for the first robot (continuum robot A). The 1-step DZNN-based algorithm and CTZND showed a similar result, but the GNN underperforms compared to all the other approaches. For the second continuum robot (continuum robot B), the original path-tracking task for the end-effector was reduced to moving from one point to the other on a surface. With the 3-step DZNN-based control, the continuum robot can reach the goal point successfully while the position error decreases exponentially. [28]

This approach showed a data-driven motion control algorithm, where NN was used to describe the system behavior, which was integrated into simulation and on a real robot to certify their usability for real-world problems in the field of robotics.

2) Motion Control with Robot-Manipulation: The robot manipulation approach from [29] involves autonomously grasping the object with a soft robotic hand. The soft robotic hand has the advantage due to its intrinsic compliance to mold around external items and exploit their environment. This leads to an increase in its grasping capabilities. To verify their data-driven control approach, they conducted grasping experiments on the objects placed on the table. With the information from a camera, first, the detection and classification of the object via the NN are started, and second, the NN predicts a suitable human reactive primitive from the generated dataset as explained in Section IV-B.2. During the experiment in [29], three tests of the 20 objects were conducted, which were not part of the training set of the NN. After one test, the position and orientation are changed. The grasp was classified as correct if the robot could have the object in its hand for 5 seconds. It was shown that the predicted strategy from the NN for some objects changed after setting different positions and orientations. The rate for a successful autonomous grasp of an object from the soft robotic and maintaining the object for 5 seconds was 81.1%. [29]

To conclude, the data-driven motion control algorithms for robot manipulation task [29] showed practical experiments of autonomous grasping with a NN as a classifier in a direct-data-driven manner.

3) Walking Assistance Control: Simulation and experiments were shown for the walking assistance control of a Lower Limp Exoskeleton (LLE) for hemiplegic patients. In the simulation, the weights of the ACNN converge after 6 seconds of learning, and the affected leg shows good tracking performance of the unaffected leg. Also, disturbances (white noise) are simulated to verify the robustness of the controller. It was shown that after 200 seconds, the unaffected leg returned to the same trajectory. [32]

For the experiment, the AIDER system was used, which is designed for walking assistance for hemiplegic patients. Three healthy patients were chosen to walk for 80 seconds. It is important to note that the healthy patients simulated paraplegia. At the beginning of the experiments, the DDRL uses the online system data to adapt to different subjects till it learns the optimal controller for LLE. The results show that after 15 seconds, the weights of the ACNN are (bounded) convergent since, in hardware experiments, the LLE gets the entire time affected by disturbances. Here, the DDRL approach was successful for walking assistance control of LLE for different patients. [32]

The results of the simulation and experiment of data-driven control with a NN as a replacement for the traditional system description were discussed.

C. DDDPC-Approach: Persistent excitation

As we mentioned before, keeping the persistency excitation of input is essential, and in this section, we present some simulations that illustrate the importance of this approach.

In [35], the performance of the closed-loop system with three different controllers was compared, starting from the same initial conditions. Controllers (i) and (ii) enforced the persistence of excitation (PE) by randomly sampling the input. For controller (iii), PE was enforced a priori. Controller (i) achieved the best performance by enforcing exact nonlinearity cancellation. Controller (iii) outperformed controller (ii), even though they used the same set of basis functions, suggesting that the region of attraction for controller (iii) may be larger than that of controller (ii), which can be attributed to the fact that a higher level of PE was achieved through multiple experiments.

Due to the artificial increase in the rank of the data matrix caused by measurement noise, the issue of whether the input data still exhibits persistent excitation becomes a concern. In [8], the authors use the SVD methods and choose different thresholds to filter how much information is contained in the input data. Furthermore, the author mentioned a good strategy for deciding whether to update the input data. If the current input data does not satisfy persistent excitation, previous data will be used for the following calculation.

In [8], there is a simulation example where a proposed data updating strategy was applied to data-driven predictive control of a two-link robotic arm affected by measurement noise. The method showed improved control performance compared to alternative strategies of using only initially available data or updating the data at each time step based on persistently exciting inputs.

All simulations show that checking the persistent excitation of the input data first and using corresponding strategies to adopt it will improve the final result a lot.

VI. Conclusion

The constant progress in robotics has pushed other areas to find better, more efficient, and more straightforward ways to control these complex systems. As seen in Section V, DDDPC theory has been used to develop simulations and some practical examples of how to bypass the model of a system entirely and achieve reliable results. However, total dependence on data means ensuring the data is reliable, robust, and safe. Furthermore, these assurances must be extended to the methods that use this data, which explains why simulations are still the primary implementation of DDDPC while practical experiments are still done on a small scale. Researchers must first prove the ability to safely control complex robotic systems with data before those methods can be extended freely to real-life interactions. From our point of view, the research in the DDDPC domain is in an early stage since insufficient methods could be found that show their ability to solve real-world robotic problems in a DDDPC manner, with practical results confirming their capabilities.

Nonetheless, in the future, we expect to see a more robust theoretical baseline for dealing with noisy data and guarantee its quality for the control methods. Furthermore, more work is needed to improve the highly complex computation when dealing with nonlinear systems requiring massive amounts of data. This review paper gives an overview of the theory and foundations of DDDPC as well as its most promising implementations.

References

[1] J. Wu, Z. Jin, A. Liu, L. Yu, and F. Yang, "A survey of learning-based control of robotic visual servoing systems," *Journal of the Franklin Institute*, vol. 359, no. 1, pp. 556–577, 2022.

[2] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013. Data-based Control, Decision, Scheduling and Fault Diagnostics.

[3] I. Markovsky, L. Huang, and F. Dörfler, "Data-driven control based on the behavioral approach: From theory to applications in power systems," *IEEE Control Syst.*, 2022.

[4] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," *IEEE International Conference on Robotics and Automation*, 2017.

[5] K. Y. Chee, T. Z. Jiahao, and M. A. Hsieh, "Knode-mpc: A knowledge-based data-driven predictive control framework for aerial robots," *IEEE Robotics and Automation Letters*, 2021.

[6] C.-Y. Wei, M. J. Jahromi, H. Luo, H. Sharma, and R. Jain, "Model-free reinforcement learning in infinite-horizon average-reward markov decision processes," in *International conference on machine learning*, pp. 10170–10180, PMLR, 2020.

[7] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annual Reviews in Control*, vol. 52, pp. 42–64, 2021.

[8] J. Teutsch, S. Ellmaier, S. Kerz, D. Wollherr, and M. Leibold, "An online adaptation strategy for direct data-driven control," *arXiv preprint arXiv:2304.03386*, 2023.

[9] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi, "Willems' fundamental lemma for state-space systems and its extension to multiple datasets," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 602–607, 2020.

[10] B. Nortmann and T. Mylvaganam, "Direct data-driven control of linear time-varying systems," *arXiv preprint arXiv:2111.02342*, 2021.

[11] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[12] V. Krishnan and F. Pasqualetti, "On direct vs indirect data-driven predictive control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 736–741, 2021.

[13] B. Nortmann and T. Mylvaganam, "Data-driven control of linear time-varying systems," pp. 3939–3944, 12 2020.

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

[15] Z. Hou, R. Chi, and H. Gao, "An overview of dynamic-linearization-based data-driven control and applications," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4076–4090, 2017.

[16] R. Cao, H. Zhou, Z. Hou, and K. Jia, "Data-driven model-free direct adaptive generalized predictive control for linear motor based on cspace," in *Proceedings of 2011 International Conference on Electronic Mechanical Engineering and Information Technology*, vol. 1, pp. 404–408, 2011.

[17] M. Mejari and A. Gupta, "Direct data-driven computation of polytopic robust control invariant sets and state-feedback controllers," 2023.

[18] S. Yahagi and I. Kajiwara, "Direct data-driven tuning of look-up tables for feedback control systems," *IEEE Control Systems Letters*, 2022.

[19] M. Suzuki and O. Kaneko, "Direct data-driven control approach of reference shaping for two degree of freedom control," *SICE Journal of Control, Measurement, and System Integration*, vol. 15, no. 1, pp. 77–85, 2022.

[20] C. Verhoek, P. J. W. Koelewijn, S. Haesaert, and R. Tóth, "Direct data-driven state-feedback control of general nonlinear systems," 2023.

[21] M. L. Scarpa, B. Nortmann, K. Y. Pettersen, and T. Mylvaganam, "Data-driven control of planar snake robot locomotion," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 1823–1828, 2022.

[22] X. Fu and K. You, "Direct data-driven stabilization of nonlinear affine systems via the koopman operator," *IEEE Conference on Decision and Control*, 2022.

[23] L. Shi and K. Karydis, "Enhancement for robustness of koopman operator-based data-driven mobile robotic systems," *IEEE International Conference on Robotics and Automation*, 2021.

[24] H. Yu, Q. Tan, L. Han, X. Dong, Q. Li, and Z. Ren, "Online data-driven model predictive control in variable noise environment based on neural network and gaussian process regression," *ACM Cloud and Autonomic Computing Conference*, 2022.

[25] A. Prieto, B. Prieto, E. M. Ortigosa, E. Ros, F. Pelayo, J. Ortega, and I. Rojas, "Neural networks: An overview of early research, current frameworks and new challenges," *Neurocomputing*, vol. 214, pp. 242–268, 2016.

[26] J. Wong, V. Makoviychuk, A. Anandkumar, and Y. Zhu, "Oscar: Data-driven operational space control for adaptive and robust robot manipulation," *IEEE International Conference on Robotics and Automation*, 2021.

[27] M. Lutter, K. Listmann, and J. Peters, "Deep lagrangian networks for end-to-end learning of energy-based control for under-actuated systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7718–7725, Nov 2019.

[28] N. Tan, P. Yu, Z. Zhong, and Y. Zhang, "Data-driven control for continuum robots based on discrete zeroing neural networks," *IEEE Transactions on Industrial Informatics*, 2022.

[29] C. D. Santina, V. Arapi, G. Averta, F. Damiani, G. Fiore, A. Settimi, M. G. Catalano, D. Bacciu, A. Bicchi, and M. Bianchi, "Learning from humans how to grasp: A data-driven architecture for autonomous grasping with anthropomorphic soft hands," *IEEE Robotics and Automation Letters*, 2019.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[31] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016.

[32] Z. Peng, R. Luo, R. Huang, J. Hu, K. Shi, H. Cheng, and B. K. Ghosh, "Data-driven reinforcement learning for walking assistance control of a lower limb exoskeleton with hemiplegic patients," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9065–9071, May 2020.

[33] C. D. Persis and P. Tesi, "On persistency of excitation and formulas for data-driven control," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 873–878, 2019.

[34] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi, "Willems' fundamental lemma for state-space systems and its extension to multiple datasets," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 602–607, 2020.

[35] M. Alsalti, V. G. Lopez, and M. A. Müller, "On the design of persistently exciting inputs for data-driven control of linear and nonlinear systems," *IEEE Control Systems Letters*, 2023.

[36] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Linear tracking mpc for nonlinear systems—part ii: The data-driven case," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4406–4421, 2022.

[37] J. Coulson, H. van Waarde, and F. Dörfler, "Robust fundamental lemma for data-driven control," *arXiv preprint arXiv:2205.06636*, 2022.

[38] N. Tan, M. Zhang, and P. Yu, "Controlling robot manipulators using gradient-based recursive neural networks," in *2020 International Symposium on Autonomous Systems (ISAS)*, pp. 207–211, Dec 2020.

[39] N. Tan, P. Yu, X. Zhang, and T. Wang, "Model-free motion control of continuum robots based on a zeroing neurodynamic approach," *Neural Networks*, vol. 133, pp. 21–31, 2021.

[40] C. Della Santina, A. Bicchi, and D. Rus, "On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control," *IEEE Robotics and Automation Letters*, vol. 5, pp. 1001–1008, April 2020.