Fakultät für Informatik
Lehrstuhl für Echtzeitsysteme und Robotik

Data-Driven Control for Large-Scale Systems: A Survey

Dominik M. Weber

Seminar Course *Cyber Physical Systems* WS 2025/2026

Advisor:	Yongkuan Zhang
Supervisor:	Prof. Dr.-Ing. Matthias Althoff
Submission:	01. January 2026

Data-Driven Control for Large-Scale Systems: A Survey

Dominik M. Weber
Technical University of Munich
Email: dominik.m.weber@tum.de

***Abstract*—Large-scale dynamical systems pose fundamental challenges for classical control due to high dimensionality and modeling complexity. Data-driven control offers an alternative by using measured data instead of relying on first-principles models. This survey reviews data-driven control methods for large-scale systems, covering both data-driven modeling techniques and direct model-free approaches. The control architectures of decentralised, distributed, and hierarchical control are discussed with a focus on enabling scalability to large-scale systems. The survey highlights the state of the art in applying data-driven approaches to large-scale systems, including open research challenges.**

I. Introduction

Large-scale dynamical systems arise in a wide range of application domains, including power networks [1], transportation systems [2], industrial processes [3], and multi-agent robotic systems [4]. Their defining characteristics, including high dimensionality and strong interconnections, often render classical centralised control approaches difficult to model and computationally prohibitive. As a result, relying on first-principles models for control of these systems becomes increasingly costly and error-prone as complexity scales [5].

Data-driven control has emerged as a promising solution to some of these limitations. Instead of relying on explicit physical modeling, data-driven approaches exploit measured system trajectories to infer models or synthesise controllers directly. This shift has enabled control designs for systems where traditional modeling is infeasible or where operating conditions change over time [6]. However, the applicability of data-driven methods to large-scale systems is not solely determined by the choice of control framework. Instead, maintaining computational tractability as system size increases hinges on how systems are decomposed, how information is exchanged, and how control decisions are coordinated across subsystems [7].

While the literature on data-driven control is extensive, its interplay with large-scale system control scenarios has received less focus. Research on large-scale control architectures often assumes access to accurate models and does not fully account for model uncertainties and difficulties for obtaining such models in real-world settings [8–10]. This survey aims to provide a structured overview of data-driven control for large-scale systems, emphasising how modeling choices, control paradigms and architectural design jointly determine scalability. It highlights scalable methods of both indirect and direct data-driven control and discusses centralised and non-centralised architectures.

The remainder of the survey is organised as follows. Section II provides a brief background on data-driven control, large-scale systems and safety filters, the central technique for providing formal guarantees for data-driven controllers. Following is an introduction to direct and indirect data-driven control techniques, forming the basis for large-scale deployment in non-centralised architectures. Section III reviews data-driven modeling techniques relevant to large-scale systems, enabling system decomposition and dimensionality reduction. Section IV covers direct, model-free control approaches and their extensions to large-scale and distributed settings. Section V discusses decentralised, distributed, and hierarchical control architectures, analysing how architectural choices affect scalability, coordination, and performance. Safety and robustness mechanisms are introduced, focusing on aspects necessitated by the system's scale. Section VI concludes with a brief discussion of open challenges and directions for future research.

II. Background

A. Data-Driven Control

A fundamental delimination in data-driven control can be made between direct and indirect approaches. In indirect methods, a model is learned from data - a process termed system identification (SI) - and, as in traditional control, a controller built upon it. In direct data-driven control, the dynamics of the system are treated as a black box and the controller is directly built from the data [5].

Data-driven approaches provide a range of benefits compared to traditional model-based ones. System identification, the basis for indirect data-driven methods, has the key advantage of being generally applicable and to a large degree automated, whereas a model-based technique requires expert knowledge for deriving the model from first principles, the specific result then being domain-specific. SI further provides the practitioner with tools to fine-tune the accuracy required, as complexity tends to rise with more accurate models [6].

Following is a brief overview of a selection of the most fundamental definitions and theorems in data-driven control, which lay the basis for many of the concepts discussed in this survey. As a comprehensive treatment of all mentioned terms is beyond the scope of this survey, the reader is instead referred to the corresponding cited literature.

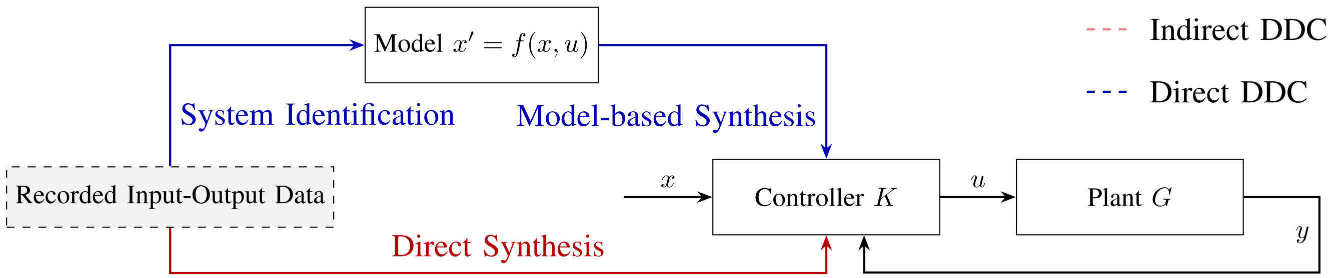

Fig. 1. Direct vs. indirect data-driven control

Hankel Matrix: A *Hankel matrix* $\mathcal{H}^i_{[N,T]}(x)$ derived from a dataset x with depth N and length T is defined as [11]:

$$\begin{pmatrix} x(i) & x(i+1) & \dots & x(i+T-1) \\ x(i+1) & x(i+2) & \dots & x(i+T) \\ \vdots & \vdots & \vdots & \vdots \\ x(i+N-1) & x(i+N) & \dots & x(i+T+N-2) \end{pmatrix} \quad (1)$$

Willems' Fundamental Lemma: A signal $u = [u_1, \dots, u_T]$ is said to be *persistently exciting of order* L if the Hankel matrix $\mathcal{H}_{[L,T]}(u)$ has full row rank [12]. Based on this notion, *Willems' Fundamental Lemma* [13] is built. As presented in [14] for the linear time invariant system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (2)$$

with (A, B) controllable, the input/state/output trajectory $(u_{[0,T-1]}, x_{[0,T-1]}, y_{[0,T-1]})$ with the stacked Hankel matrices

$$\begin{pmatrix} \mathcal{H}_{[L,T]}\left(u_{[0,T-1]}\right) \\ \mathcal{H}_{[L,T]}\left(y_{[0,T-1]}\right) \end{pmatrix} = \begin{pmatrix} u(0) & \dots & u(T-L) \\ \vdots & \vdots & \vdots \\ u(L-1) & \dots & u(T-1) \\ y(0) & \dots & y(T-L) \\ \vdots & \vdots & \vdots \\ y(L-1) & \dots & y(T-1) \end{pmatrix} \quad (3)$$

assuming the input $u_{[0,T-1]}$ is persistently exciting of order $n + L$, the following two statements hold: Firstly, any input/output trajectory $(\bar{u}_{[0,L-1]}, \bar{y}_{[0,L-1]})$ is a trajectory of the system in eq. (2) if and only if, for some real vector g,

$$\begin{pmatrix} \bar{u}_{[0,L-1]} \\ \bar{y}_{[0,L-1]} \end{pmatrix} = \begin{pmatrix} \mathcal{H}_{[L,T]}\left(u_{[0,T-1]}\right) \\ \mathcal{H}_{[L,T]}\left(y_{[0,T-1]}\right) \end{pmatrix} g \quad (4)$$

Secondly, the following matrix has full row rank:

$$\begin{pmatrix} \mathcal{H}_{[1,T]}(x_{[0,T-L]}) \\ \mathcal{H}_{[L,T]}(u_{[0,T-1]}) \end{pmatrix} = \begin{pmatrix} x(0) & \dots & x(T-L) \\ u(0) & \dots & u(T-L) \\ \vdots & \vdots & \vdots \\ u(L-1) & \dots & u(T-1) \end{pmatrix} \quad (5)$$

B. Large-Scale Systems

Large-scale systems do not have a single clear definition. The term is rather used whenever a system grows intractable to control with traditional strategies. It is further often used synonymously with complex system, a system consisting of multiple interacting units that exhibits emergent collective behaviour which is not the simple summation of its parts [15]. Application domains of large-scale system control are areas such as power generation and distribution, traffic and water networks, and industrial system processes, such as heating ventilation or supply chain management systems [16].

The central problem in controlling such systems is the scalability requirements they impose upon the controller, often reaching thousands of interacting entities [15]. Traditional first-principle-based control of such systems can be intractable to formulate and compute, while being sensitive to uncertainty [17], motivating the usage of data-driven methods. Generally, techniques used to handle control of large-scale and complex systems include reducing the model to a more tractable size through model order reduction methods, utilising non-centralised control architectures and including communication protocols and interconnects between individual subsystems into the control model.

Model Order Reduction methods tackle the problem of reducing the complexity of large-scale systems. This technique attempts to generate reduced models reproducing the input-output behaviour of large-scale systems accurately enough for the desired use case. Traditional model reduction methods fall into two categories: Singular Value Decomposition (SVD)-based and moment matching methods. SVD-based techniques are related to the eponymous singular value decomposition. Their core idea is the computation of controllability and observability Gramians, eliminating hard-to-reach and -observe states [18]. Their advantages include the preservation of stability and a computable error bound. While work is being done to make them more efficient using approximations [19, 20], their resource requirements remain their main drawback [18]. A popular method in this class of model order reduction techniques is Balanced Truncation (BT) [21]. Moment matching methods consider the problem of matching the coefficients of power series expansions of the transfer function at selected points, reducing model reduction to rational interpolation. They tend to be cheaper to compute than SVD-based methods, but do not provide as strong of theoretical guarantees, the preservation of properties depending on factors such as the choice of expansion points [22]. A prominent example is Krylov subspace methods [23].

The second key approach taken to enable the control of large-scale systems is the decomposition of the centralised control problem into smaller, more manageable control tasks distributed to sub-controllers. A centralised controller often does not suffice for effectively controlling large scale systems, the necessity of multiple controllers even being part of some definitions of large-scale systems [16]. Depending on the interaction between the controllers, such approaches can be categorised into decentralised, distributed and hierarchical control architectures, in addition to hybrids thereof [16].

Control of large-scale systems can roughly be categorised into three distinct levels: node control, edge control and structural control [15]. Node control concerns itself with the immediate determination of dynamics of a subset of agents at the lowest, microscopic view. Edge control works by dynamically adjusting the communication protocol between the individual agents. In structural control, the network topology can be reshaped in order to guide behaviour at the macroscopic level. Most works are concerned with node control, as this resembles the traditional single, small-system control the closest. While not all layers might be controllable for a specific use-case, such as the network topology being fixed and unchangeable, the best results are usually achieved when considering all levels [15].

C. Safety filters

The inherent complexity of large-scale systems not only challenges tractability, but also exacerbates safety and certification concerns. As data-driven controllers replace interpretable physics-based models, formal guarantees must often be reintroduced. This section serves as an introduction to safety filters, a popular approach for achieving formal guarantees in data-driven large-scale systems [24–26].

A system $\dot{x}(t) = f(x(t), u(t))$ is called safe if

$$\forall t \in \mathbb{R}_{\geq 0} x(t) \in \mathcal{X} \wedge u(t) \in \mathcal{U} \tag{6}$$

where $\mathcal{X}$ and $\mathcal{U}$ are the state constraint set and the input constraint set, respectively [27].

Unlike traditional model-based designs, learning-driven controllers must operate under epistemic uncertainty arising from limited data, unmodeled dynamics, and distributional drift. Consequently, a major strand of recent work focuses on safety filters. These filters are supervisory mechanisms that ensure system trajectories remain within a certified safe set, even when the underlying controller or policy is learned. Formally, a safety filter is a function $\kappa : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathcal{U}$ which transforms the control signal ($u(t) = \kappa(x(t), u_{contr}(t))$) determined by the controller u_{contr} such that safety constraints are respected, and the modification is minimal, i.e. minimizing

$$\int_0^{\infty} ||\kappa(x(t), u_{contr}(t)) - u_{contr}(t)|| dt \tag{7}$$

Among these, three major methodological families dominate current research [27]: Control Barrier Function (CBF) approaches, Hamilton-Jacobi reachability-based methods, and Predictive control formulations.

For a known control affine system $\dot{x} = f(x) + g(t)u$, a Control Barrier Function can be defined as follows [28]: a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$ is a CBF if there exists an extended class K_∞ function α:

$$\sup_{u \in \mathcal{U}} \dot{h}(x, u) \geq -\alpha(h(x)) \tag{8}$$

for all $x \in \mathcal{X}$ where $\dot{h}(x, u) = \nabla h(x) f(x) + \nabla h(x) g(x) u$.

The key idea is that for the safety set $S = \{x \in \mathcal{X} | \exists u \in \mathcal{U} \wedge h(x) \geq 0\}$, any Lipschitz continuous controller $u(t)$ is safe [29]. While CBFs traditionally rely on explicit system models, recent research has proven the viability of learning them from data as well [30, 31].

Hamilton-Jacobi reachability analysis computes the backward reachable tube, a collection of all initial states with trajectories eventually reaching a target set even under the worst-case disturbance. For safety, this target set can be the set of all unsafe states, the backward reachable tube then representing all potentially unsafe, to-be-avoided states [32]. While this approach is easily generalisable to many applications, scalability concerns can be a limiting factor [33].

Predictive safety filters enforce safety constraints by solving a constrained optimisation problem that minimally corrects the nominal control input. At each time step, the filter predicts the future system evolution over a finite horizon and ensures that all reachable states remain within the safety set. A key concept is the use of backup trajectories, feasible fallback control sequences that guarantee the system can return to a terminal safe set even if the nominal policy becomes unsafe. By verifying the existence of such a backup trajectory before applying the proposed control, predictive safety filters ensure recursive feasibility and forward invariance of the safety set [34].

III. Data-Driven Modeling

To enable indirect data-driven control techniques, a model of the to be controlled system needs to be identified. This section outlines data-driven modeling techniques suitable for large-scale systems, beginning with system identification, where collected data is used to construct a model of the system's underlying dynamics. Given the scale and structural complexity of modern interconnected systems, topology identification and decomposition is subsequently addressed, revealing the system's interaction structure and partitioning it into manageable subsystems. Finally, for components exhibiting significant nonlinear behavior, Koopman-based lifting techniques are discussed, enabling the approximation of nonlinear dynamics through finite-dimensional linear operators.

A. System Identification

As large-scale systems tend to produce system models that are unwieldy to handle, data-driven model order reduction techniques receive special attention [35]. Data-driven model order reduction methods rely only on measured trajectories, frequency response data, or impulse responses and produce a reduced representation of the system's dynamics.

1) Interpolatory approaches: A widely used approach is data-driven interpolatory model reduction. One such method, the Loewner framework, constructs a reduced linear model from transfer-function data or sampled impulse responses [36]. The main element of the Loewner framework is the Loewner Matrix. For two lists of complex number pairs u_i, v_i and μ_j, λ_j, it is defined as follows [18]:

$$\begin{pmatrix} \frac{v_1-w_1}{\mu_1-\lambda_1} & \cdots & \frac{v_1-w_k}{\mu_1-\lambda_k} \\ \cdots & \cdots & \cdots \\ \frac{v_q-w_1}{\mu_q-\lambda_1} & \cdots & \frac{v_q-w_k}{\mu_q-\lambda_k} \end{pmatrix}$$

The rank of this Loewner matrix then contains information about the minimal admissible complexity of the solutions of the interpolation. The shifted Loewner matrix L_σ has the entries $L_{\sigma ij} = \frac{\mu_j v_j - \lambda_i w_i}{\mu_j - \lambda_i}$. The key insight is that for a linear system, these matrices are directly related to the system's internal state-space matrices without needing to know them explicitly [37]. The singular values of the Loewner matrix are used to determine the numerical rank of the system, which suggests the appropriate order of the reduced model. A sharp drop-off in the singular values indicates a good candidate for the reduced model's dimension. By selecting a small number of the largest singular values, one can capture the most important dynamics of the original system. The reduced-order model's state-space matrices are then directly computed from the projected Loewner matrices and the data vectors [18]. While the Loewner framework has been successfully employed in large-scale system scenarios [36], the $\mathcal{O}(n^3)$ complexity of SVD and its sensitivity to noise [38] necessitates the usage of advanced implementation techniques, such as using a sequence of 1-dimensional Loewner matrices for the computation of the null-space of an n-dimensional one, as proposed in [39], reducing the time complexity to about $\mathcal{O}(n^{1.4})$.

Another interpolatory model order reduction technique is the AAA algorithm [40], which builds rational approximants adaptively by selecting interpolation points based on approximation error. This method is attractive when only frequency response data is available, as it balances accuracy and model complexity with minimal user intervention.

2) Balanced truncation methods: Balanced truncation [21] is a widely successfully employed model order reduction technique for large systems [41], traditionally relying on controllability and observability Gramians and hence access to system matrices. A data-driven reformulation is presented in [42]. The fundamental insight driving this development is that BT does not use the Gramians directly, but rather their product, which can be approximated from data. The results of the numerical BT method can get arbitrarily close to those of classical BT, depending on how many computational resources are allocated to the numerical quadrature.

As in the Lowener framework, the SVD involved grows computationally infeasible for matrices common in large-scale systems. Using extended Krylov subspace methods, a low-rank approximation to the corresponding Sylvester equations can be generated, greatly improving efficiency [43]. Another approach is to utilise the idea of the low-rank Alternating Direction Implicit method [44], a popular method for handling the computational complexity in traditional BT settings [45], and reformulate it for data-driven BT, reducing the model using transfer function samples from the right half of the s-plane [46].

3) Modal decomposition methods: Proper orthogonal decomposition is one of the most widely used dimensionality-reduction techniques in dynamical systems. The key idea is to approximate system trajectories by projecting them onto a low-dimensional basis obtained directly from data. For a dataset $X = [x_1 x_2 ... x_m], x_i \in \mathbb{R}^n$, proper orthogonal decomposition seeks an orthonormal basis $U_r \in \mathbb{R}^{n \times r}$ that minimizes the projection error

$$\min_{U_r^T U_r = I_r} ||X - U_r U_r^T X||_F^2 \tag{9}$$

This optimisation problem is solved by computing the singular value decomposition of the snapshot matrix X. The dominant singular vectors define the reduced basis, and the singular values quantify the energy captured by each mode [47].

A closely related algorithm is Dynamic Mode Decomposition (DMD). Given snapshot pairs $S^- = [x_1, ..., x_{m-1}]$ and $S^+ = [x_2, ..., x_m]$, DMD approximates the system by finding a matrix A such that

$$x_{k+1} \approx A x_k \tag{10}$$

The goal is to determine a low-rank subspace of A and through it represent system dynamics. One first performs SVD on $S^- = U \Sigma V^T$, truncating U and V to the first r columns and selecting the first r singular values from Σ. What value to use for r depends on accuracy requirements, though a typical choice is to have it retain a specific fraction of information, such that

$$r = \arg\min_j \frac{\sum_{i=1}^j \sigma_i}{\sum_{i=1}^n \sigma_i} < \tau \tag{11}$$

One can then reconstruct $A_r = U_r S^+ V_r \Sigma_r^{-1}$. Using the cheap eigen-decomposition $A_r W = \Lambda W$, one can compute the DMD modes $\Psi = U_r W$, which contain the essential information about the system's evolution and can be used to approximately predict the system's state at any point in time [48].

A drawback of DMD is its sensitivity to sensor noise, which stems from its nature as effectively a least-squares algorithm, resting on the assumption that the independent variable is noise-free [49]. Making DMD explicitly aware of noise, e.g. using a Kalman filter [50] or a total least-squares approach [51], solves this problem to some degree [52], though potentially at the cost of computational stability [53]. The scalability of DMD, dominated by the SVD, can be improved via approaches such as working on compressed data [54], exploiting multi-scale dynamics via explicit introduction of multi-resolution DMD [55], or embedding DMD into a probabilistic framework to get a randomised DMD algorithm [56].

B. Topology Identification and Decomposition

An aspect of system identification central for large-scale systems is topology identification. Whereas classical SI seeks simply to determine compact dynamic representations, topology identification aims to discover structural information about the subcomponents of the system. In large-scale systems, the goal of modeling is often to identify a sparse structure therein. This lays the necessary basis for popular decentralised and distributed control techniques [57]. Formally, given a set of variables V, a directed graph $\mathcal{G} = (V, \mathcal{E})$ is to be determined from data, such that $(v, w) \in \mathcal{E}$ if v has a direct causal effect on w [5].

The system is hence not considered as a single, coherent object anymore, but rather as a collection of different interacting subsystems. The two primary decomposition methodologies are physical and numerical. In the former, the subsystems are defined by the physical components of the system, such as individual robots in a swarm [58]. Numerical decomposition, in contrast, imposes such structure based on computational reasons, partitioning the large-scale system into manageable subsystems [57]. The interaction topology of the system is essential to consider in such approaches, as it provides information on the dynamical couplings, allowing strongly connected components to remain within one subsystem, the motivation being that such modular partitioning tends to improve performance by minimizing inter-subsystem interaction [59].

A widely adopted approach is the Sparse Identification of Nonlinear Dynamics (SINDy) framework [60]. It constructs a library of candidate nonlinear functions $\Theta(X)$ and identifies governing interactions via sparse regression based on the time-series data $X = [x_1, ..., x_n]$:

$$\dot{X} = \Theta(X)\Xi \tag{12}$$

$$\Xi = \arg\min_{\Xi} \frac{1}{2}||\dot{X} - \Theta(X)\Xi||_2^2 + \lambda||\Xi||_1 \tag{13}$$

The resulting sparsity pattern in Ξ directly defines the data-driven interconnection topology. SINDyG [61], an extension of the base SINDy method, incorporates the network structure into sparse regression, directly identifying model parameters explaining underlying network dynamics, allowing the capturing of small changes in the emergent behaviour.

Another family of approaches uses causal discovery from time-series, generalizing Granger causality [62] to nonlinear and high-dimensional dynamics. Methods such as PCMCI [63] recover directed interaction graphs by statistically testing predictive dependencies in data, requiring no explicit model assumptions.

Once the interaction topology is estimated, data-driven decomposition partitions the overall network into weakly coupled modules. This step is essential for scalability, as it limits the dimensionality of each identification and control problem. Practical strategies include clustering based on finding highly connected subgraphs with balanced number of internal and external connections [57] and spectral clustering on dynamic interaction graphs [64].

C. Koopman lifting for Nonlinear Subsystems

For subsystems with strongly nonlinear dynamics, linear models identified via traditional system identification techniques are often inadequate [65]. A recent trend in control of nonlinear systems is the usage of methods based on the Koopman operator [66–69]. Koopman-based methods lift the state to a space of observables where the nonlinear evolution can be captured exactly or closely approximated by a linear operator. This enables the use of linear control and estimation tools on inherently nonlinear subsystems while retaining the structure and couplings of the full system. For large-scale systems, this approach allows scalable controller design, bridging the gap between nonlinear subsystem behavior and the computational advantages of linear control.

Formally, the Koopman operator $\mathcal{K}_f : \mathcal{F} \to \mathcal{F}$ associated with the state transition map $f : X \to X$ and the Banach space $\mathcal{F}$ of observables $g : X \to \mathbf{C}$ is defined through the composition $\mathcal{K}_f g = g \circ f, \forall g \in \mathcal{F}$. The two core properties making the Koopman operator an exceedingly useful tool are its globality and linearity [70]. The Koopman operator provides an exact linear approximation to nonlinear systems that is globally valid, instead of only around a point or trajectory. The main drawback, however, is that it can be infinite-dimensional [71].

A Koopman-invariant subspace is the span of a set of m basis observable functions y_i if all functions g lying in this subspace, i.e. $g = \sum_{k=1}^{m} \alpha_k y_k$ remain inside this subspace upon application of the Koopman operator, there being a linear combination of the basis functions representing $\mathcal{K}g$, $\mathcal{K}g = \beta_1 y_1 + \beta_2 y_2 + ... + \beta_m y_m$ [72]. On this subspace, a finite-dimensional linear operator K can be obtained, restricting the Koopman operator $\mathcal{K}$ and describing the linear evolution of lifted observables [71]. When applying the Koopman operator to control scenarios, a Koopman-invariant subspace containing the original state variables x_i is desired. This can be an impossible task for some systems, for example, a system with multiple fixed points, as all finite-dimensional linear systems have a single fixed point [72].

The data-driven determination of the Koopman operator's spectral properties usually involves projecting the infinite-dimensional operator onto a finite-dimensional subspace of observables [73]. This finite-dimensional approximation is obtained directly from data, hence no intermediate high-dimensional system to be reduced afterwards is identified. DMD [74] is a popular technique, in large part due to its ease of implementation [75], extensibility and interpretability [76]. The procedure begins by collecting a time-series of state snapshots and organising them into two matrices, X and Y, representing paired snapshots of the system's past and future states, respectively. The core objective is to determine the best-fit linear operator A such that $Y \approx AX$. To ensure robustness and manage high-dimensional data, the method utilises the SVD of the state matrix $X \approx U_r \Sigma_r V_r^*$ to project the dynamics onto a rank-r subspace [77]. The low-rank approximation of the Koopman operator is then calculated as $\tilde{A} = U_r^* Y V_r \Sigma_r^{-1}$ [74].

Another widespread [78] method for linear functions is tICA [79]. The methods of eDMD [80], VAC [81], KernelDMD [82] and kernel tICA [83] extend the function dictionary to non-linear functions [73]. The latter two kernel techniques resort to being infinite-dimensional, trading off computational efficiency for theoretical guarantees [84], though approximate kernel methods are a topic of active research [73, 85, 86].

Despite their theoretical appeal, Koopman-based approaches face well-documented limitations in large-scale settings. First, finite-dimensional Koopman-invariant subspaces rarely exist for nonlinear systems of practical interest, causing lifted linear models to incur rapidly growing approximation errors [72]. Second, the observable dictionaries required for expressive embeddings scale combinatorially with system complexity, making both eDMD-type methods and dictionary learning computationally prohibitive at large scale [70]. Kernelised variants partially avoid explicit feature construction but introduce $\mathcal{O}(N^2)$ to $\mathcal{O}(N^3)$ computational cost in the number of snapshots, which quickly becomes intractable without sketching or other aggressive approximations [73]. As a consequence, while the application of Koopman theory to ever-larger systems is currently a vibrant field of research, Koopman lifting remains viable only for moderate-size nonlinear subsystems with coherent low-dimensional structure rather than arbitrary full large-scale networks.

IV. Model-Free Control

Instead of data-driven modeling, where data approximates the underlying system to enable model-based control, an orthogonal strand of research eliminates explicit modeling altogether. Such model-free direct data-driven control paradigms, chiefly among them Data-enabled Predictive Control (DeePC) and Reinforcement Learning, have received much attention lately.

A. Data-enabled Predictive Control

Data-enabled Predictive Control [12] is a model-free control framework that extends the principles of Model Predictive Control (MPC) to a purely data-driven setting. It eliminates the need for explicit system identification by leveraging Willems' Fundamental Lemma [13] from behavioural systems theory, which states that all trajectories of a controllable linear system can be represented as linear combinations of a single persistently exciting trajectory.

Given a persistently exciting input-output dataset, DeePC constructs Hankel matrices of past inputs and outputs, then formulates an optimisation problem that directly predicts future trajectories and computes optimal control actions without identifying a model. As described in [12], the DeePC works as follows: Taking as input a reference trajectory r, past input-output data (u_{ini}, y_{ini}), constraint sets U, Y and performance matrices Q amd R, the first step is to solve for g^*:

$$\min_{g,u,y} \sum_{k=0}^{N-1} ||y_k - r_{t+k}||_Q^2 + ||u_k||_R^2 \tag{14}$$

subject to

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix}, \forall k \in [N-1] : u_k \in U, y_k \in Y \tag{15}$$

The optimal control sequence is subsequently calculated via $u^* = U_f g^*$.

Since its introduction, a variety of extensions have been proposed to enhance the robustness, scalability, and practicality of DeePC. To improve noise tolerance and generalisation to unseen conditions, regularised DeePC formulations introduce penalties on the trajectory coefficients and slack variables, effectively mitigating overfitting to specific datasets [87]. Scalability can be improved greatly by adding a dimension reduction step to the traditional DeePC technique. This involves performing a low-rank approximation of the trajectory matrices via SVD, which reduces the number of decision variables in the underlying optimization problem while maintaining the dominant system dynamics [88].

Its applicability to large-scale systems is primarily driven through distributed and hierarchical DeePC algorithm formulations, decomposing the global DeePC optimisation into coupled subproblems solved locally with limited coordination, improving scalability and communication efficiency [89, 90].

The extension of DeePC to the control of nonlinear systems has been a focus of recent reserach, primarily through the usage of kernel functions [91, 92]. While these works have considered computational efficiency, real-time applicability remains limited, [91] incurring a slowdown by a factor of almost ten over the original DeePC implementation. In the context of large-scale systems, these methods should hence be reserved for the control of strongly nonlinear subsystems.

Recent works further address online computational efficiency through recursive and low-rank DeePC updates, allowing adaptive, real-time control under streaming data constraints [93]. The DeePC formulation is transformed into a low-dimensional form via SVD. Then, a fast SVD update mechanism is provided to avoid recomputing everything from scratch every time new data arrives. This drastically improves online use viability.

DeePC-style control can be made robust, as proposed in [91], a computationally tractable reformulation of DeePC's traditional min-max optimisation problem via uncertainty sets, providing performance guarantees for the open-loop realised input/output cost. For partially known systems, hybrid data-enabled predictive control [94] presents an approach providing improved computational cost and robustness via an inclusion of model knowledge into DeePC.

Safety constraint satisfaction in DeePC has largely been achieved via hybrid control architectures [95], where a secondary controller activates when the predictive controller cannot guarantee satisfaction of output or trajectory constraints, and via data-driven reachability analysis [96]. Hybrid designs offer flexibility and can be computationally efficient, but they are not specific to DeePC. Meanwhile, reachability analysis scales poorly with system size [33], limiting the applicability of this approach to large-scale systems. Consequently, the development of safety mechanisms for DeePC that remain computationally tractable for high-dimensional systems remains an open challenge.

B. Reinforcement Learning

Reinforcement Learning (RL) has recently become a popular method for dealing with the ever-growing complexity of modern systems, providing the benefit of determining optimal control policies without necessitating detailed system models [97]. RL-based controllers require no labelled datasets and are instead guided by feedback and learn in a trial-and-error fashion, making them well-suited for large-scale systems where explicit system models are missing [98].

Another advantage of RL is its statement of the control problem as Markov decision processes (MDPs), providing great generality. They are applicable to nonlinear and stochastic dynamics, nonquadratic reward functions and even continuous states and actions through numerical function approximation techniques [99]. Weaknesses include high data requirements and long convergence times [100], hyperparameter sensitivity [101], potentially high computational costs, and often missing formal stability guarantees [100].

The goal is to find a policy $\pi : S \to \mathcal{A}$ mapping states to actions maximizing the expected cumulative reward

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \tag{16}$$

$\gamma \in [0, 1]$ denotes a factor balancing immediate against long-term goals and r encodes control objectives as scalar rewards, possibly differing across individual controllers, e.g. in hierarchical systems [102].

To enable real-time online reinforcement for large-scale systems, approximation approaches have been proposed, where the input-output history is transformed into a shorter set of data, being the basis for a lower-dimensional inferred controller than the original, decreasing the time complexity from $\mathcal{O}(n^6)$ to $\mathcal{O}(L^6)$, where n is the state vector size and $L \ll n$ represents the length of the input-output data history considered [103]. As a guideline for choosing a value for L that still results in a nearly optimal controller, the authors propose a SVD-based algorithm that determines L based on a threshold value for the singular values with a complexity of $\mathcal{O}(L^4)$.

A central distinction is to be made between online and offline RL. Traditionally, RL is viewed as an active learning process requiring interaction with the environment. This can, however, be expensive and dangerous, and must rely on a smaller set of data than could be gathered for offline datasets, due to them only having to be collected once, instead of during each run [104]. Offline RL methods, the learning from entirely pre-collected, offline data, without on-policy interaction, have been developed accordingly. Challenges to be solved with this approach include the inability of exploration, the counterfactuality of queries and distributional shift [105]. The learned policy may select actions that are poorly represented in the dataset, leading to extrapolation errors in value estimates. To mitigate this, a range of regularisation strategies have been proposed.

One range of technique centres around the idea of Conservative Q-Learning [104]. Conservative Q-Learning solves issues off-policy RL algorithms face by learning a conservative estimate of the value function, avoiding over-estimation and yielding much better performance. A tighter lower bound is achieved by Mildly Conservative Q-learning [106], actively training out-of-distribution actions by constructing them proper pseudo target values.

Offline RL has also been extended to multi-agent and distributed settings. Counterfactual Conservative Q-Learning proposes a method counterfactually determining conservative regularisation for each agent separately and subsequently combining them linearly, leading to an overall conservative estimation [107].

To achieve robustness in reinforcement learning, the maximization problem presented in eq. (16). can be reformulated, depending on where the uncertainty is assumed to stem from [108]. In the case of an unknown transition matrix P, fixed and lying within a known uncertainty set $\mathcal{U}_P$, this might look as follows:

$$\max_{\pi \in \Pi} \min_{P \in \mathcal{U}_P} \mathbb{E}_{P,\pi}\left[\gamma^t r(s_t, a_t)\right] \tag{17}$$

Since reinforcement learning approaches make random decisions in the process of exploration that can violate safety constraints and lead to unacceptable system behaviour, naive RL methods cannot be applied to many safety critical areas [109]. One approach to safe RL is to utilise Lyapunov functions determined through linear programming to guarantee safety and robust learning [110]. An alternative methodology involves adding CBFs, dropping the reward to minus infinity in safety violation situations, ensuring the learned optimal control policy respects the safety bounds [111]. These CBFs can further be learned efficiently using neural networks, as demonstrated by [112], improving this method's applicability to large-scale systems by extending the invariant property of the barrier certificate to multiple steps and integrating it into the policy optimisation in a computationally efficient manner using the Lagrangian method, achieving near-zero safety violations on high-dimensional control tasks. However, the learning phase typically needs to be carried out in simulation environments, as safety constraints violations are likely at this stage. A formal-methods-based approach presented in [113] safety constraints expressed in LTL guide the exploration. A key drawback of this, despite success in the safety dimension, is the necessity of manual formulation of these constraints [114].

V. Control Architectures for Large-Scale Systems

The scalability of data-driven control depends not only on the learning or identification methods employed, but also on the architecture through which control decisions are coordinated. The control architecture determines how information is shared and decisions are made within a large-scale system. Large-scale systems pose fundamental challenges for classical control architectures. As system dimensions and interconnections grow, centralised strategies become computationally intractable and communication-limited [7].

To address these challenges, the three principal paradigms of decentralised, distributed, and hierarchical control have evolved, each with characteristic trade-offs in scalability, performance, and robustness. Beyond this tripartite classification, modern work increasingly integrates hybrid architectures, blurring traditional boundaries.

A. Decentralised control

In decentralised control, each subsystem maintains an independent controller relying solely on local measurements and actuation. No online communication occurs among controllers, and couplings are treated as exogenous disturbances or modeled implicitly [115]. This structure yields high robustness to communication failures, straightforward scalability and is attractive for large-scale systems where global data aggregation is infeasible or undesirable due to privacy, latency, or safety concerns, yet typically sacrifices global optimality, as individual controllers do not have as much information regarding the state or output of the system as a centralised one would [116]. In some definitions of decentralised control, information exchange is allowed before and after the decision-making process, simply restricting the controllers from actively negotiating [117]. In this survey, in line with existing literature [16], decentralised control is understood to not allow for explicit communication between subsystems, whereas distributed control does, as illustrated in section V.

A natural approach to decentralised control is the extension of the popular MPC to this setting. In general, data-driven MPC can be formulated as presented in [118]. Given input-output data for the last ρ timesteps $\mathbf{z}_t = [u_{t-\rho}^T ... u_{t-1}^T y_{t-\rho}^T ... y_{t-1}^T]^T$ and denoting the T future inputs and predicted outputs as $\mathbf{u}_t = [u_t^T ... u_{t+T-1}^T]^T$ and $\hat{\mathbf{y}}_t = [\hat{y}_t^T ... \hat{y}_{t+T-1}^T]^T$, respectively, the goal is to minimize some some finite-horizon criterion $J(\mathbf{u}_t, \mathbf{y}_t)$. With reference trajectory $(\bar{\mathbf{y}}_t, \bar{\mathbf{u}}_t)$, this could take the form of the quadratic criterion

$$J(\mathbf{u}_t, \mathbf{y}_t) = ||\mathbf{y}_t - \bar{\mathbf{y}}_t||_Q^2 + ||\mathbf{u}_t - \bar{\mathbf{u}}_t||_R^2 \tag{18}$$

The finite-horizon predictive control problem is then given by the equations, provided constraint sets $\mathcal{U}$ and $\mathcal{Y}$:

$$\min_{\mathbf{u}} J(\mathbf{u}, \hat{\mathbf{y}}) \tag{19}$$

$$\hat{\mathbf{y}} = \text{Prediction}(\mathbf{z}, \mathbf{u}), \mathbf{u} \in \mathcal{U}, \hat{\mathbf{y}} \in \mathcal{Y} \tag{20}$$

For each timestep, the optimisation problem can be solved and the first optimal input component applied to the system, repeating until convergence [119].

The predictor can be either model-based or computed directly from the training data. As a simple illustration, a linear model for predicting the measurements is given by $\hat{\mathbf{y}} = \hat{\Theta}\phi(\mathbf{z}, \mathbf{u})$, $\phi(\mathbf{z}, \mathbf{u})$ being a regression vector and $\hat{\Theta}$ the vector of estimated parameters. A direct instantiation of data-driven MPC is the DeePC method presented in eq. (14) and eq. (15). Mattson et al. [118] showed that a range of direct methods is equivalent to an indirect method with a slack variable in the predictor.

To illustrate the changes in a decentralised MPC method, the linear dynamics of the the ith subsystem controller in a model-based view could be described as [120]

$$\dot{x}_i = Ax_i + Bu_i + Ex_c \tag{21}$$

where E denotes the disturbance based on the central state x_c, one of its sources being the actuation from other subcontrollers.

Chen et al. [121] propose a decentralised MPC system for nonlinear processes where neural network models are trained for each subsystem and used as prediction models. This approach resulted in improved computational cost compared to the centralised approach, while ensuring closed-loop state boundedness and convergence. Decentralised MPC can even work in application domains that tend to favour a different kind of architecture, such as microgrids [122].

The more popular approach to predictive control in decentralised architectures is the DeePC one, though. dDeePLCC [123], a decentralised formulation of DeePC, achieves better safety guarantees and smaller computational cost than a centralised formulation, while being naturally privacy preserving.

1) Robustness: Robustness in decentralised data-driven control addresses the challenges of uncertainty in subsystem interactions and disturbances acting on local dynamics. Unlike centralised architectures, decentralised controllers cannot rely on global state information, making explicit robustness formulations essential for guaranteeing stability and constraint satisfaction despite the limited information exchange between subsystems.

A central approach to robust decentralised predictive control introduces bounded uncertainty sets for each subsystem. These sets capture not only local disturbances and measurement noise, but also partially observed inter-subsystem coupling and network effects. This method has been successfully employed in the context of both predictive control [123, 124] and adaptive dynamic programming [125] in decentralised architectures.

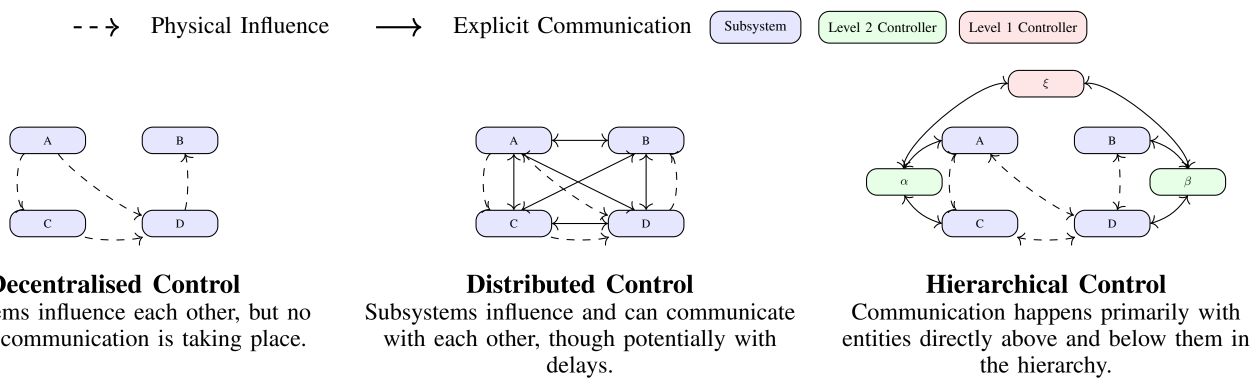

Fig. 2. An overview of non-centralised control architectures

Overall, decentralised robustness mechanisms scale well computationally with respect to number of subsystems, with approximately constant online complexity per subsystem, as they avoid global optimisation and communication entirely. However, this scalability tends not to stem from exploiting the large-scale structure of the system, but is rather a consequence of a naive decomposition of the global robustness goal to the subsystem level, yielding limited system-level robustness guarantees under strong coupling conditions.

2) Safety: The decentralised control of large-scale systems poses a special challenge for safety guarantees, as the local subsystem controllers have to ensure constraint satisfaction despite only implicitly gaining information from the other subcontrollers, no direct communication channel being available, safety hence having to be enforced locally and independently. While this tends to ensure a constant per-subsystem complexity, safety guarantees extending beyond those decomposing into local requirements can only be met under restrictive assumptions. A common method [24, 126] for achieving such guarantees is the use of neural networks projecting the proposed action onto the nearest safe one, the networks having been trained on historical data to estimate the local impact of control actions on system states.

Ultimately, research on safety for decentralised data-driven control remains limited. The lack of communication fundamentally restricts the ability to guarantee global safety properties in strongly coupled systems, making decentralised safety mechanisms most appropriate for settings with inherently local constraints.

B. Distributed control

As opposed to decentralised control, subsystems in distributed architectures explicitly share information among system components. This enhances the coordination capabilities of the agents, incurring improved scalability and robustness [116]. Factors that need to be considered in this design are communication delays and reliability.

A general formulation of distributed control for a controller i would look as follows, where each controller uses its own state and state information shared by immediate neighbours $\mathcal{N}_i$, potentially transformed through a function ξ:

$$u_i = \kappa_i(x_i, \{\xi(x_j) \mid j \in \mathcal{N}_i\}) \tag{22}$$

1) Distributed Model Predictive Control (DMPC): A family of techniques having been researched deeply for decades [127] is the collection of Distributed Model Predictive Control methods. In a DMPC setting, each subsystem solves a local MPC and exchanges certain pieces of information, which is considered in addition to dynamics, constraints and objectives. DMPC systems can be approached from a bottom-up or a top-down perspective: either a collection of autonomous systems is considered, introducing communication for coordination purposes, or from the view of a monolithic system being decomposed into subsystems, coordinated under constraints of communication and processing power [128].

In non-iterative approaches, agents coordinate only once per sampling instant to solve their optimisation problems, providing simple and fast coordination, albeit at the cost of sub-optimal performance [129]. Iterative DMPC strategies aim to overcome the sub-optimality of non-iterative methods by repeatedly exchanging information until a consensus or near-optimal solution is reached [130]. To realise this efficiently, decomposition techniques such as the alternating direction method of multipliers [131], the dual fast gradient method [130] and Nesterov-accelerated gradient [132] approaches have been used successfully.

For computational and operational efficiency reasons, constraining the communication of subsystems to a local neighbourhood can be advantageous. DLMPC [133] and the corresponding data-driven formulation D^3LMPC [134] use the system level synthesis framework [135] to generate a distributed model-predictive controller, which shares information only locally, the neighbourhood sizes being adjustable. The system level synthesis framework parameterises the entire closed-loop system using two transfer matrices called system responses $\{\Phi_x, \Phi_u\}$, such that

$$\begin{pmatrix} x \\ u \end{pmatrix} = \begin{pmatrix} \Phi_x \\ \Phi_u \end{pmatrix} w \tag{23}$$

w being the exogenous disturbance.

The data-driven formulation uses Hankel matrices to replace to closed-form system responses [136]. DLMPC achieves the restriction of neighbourhood size, the maximum number of hops in the communication graph a subsystem can communicate with, through a locality constraint on the system response. The data-driven DLMPC provides the same guarantees as the standard DLMPC, namely recursive feasibility, stability and convergence for both nominal and robust settings [137].

2) Multi-Agent RL: In Multi-Agent RL, each agent controls a subsystem and must learn policies in a partially observable environment while interacting with other agents. As in model-based methods, centralised control via RL can grow intractable. In a multi-agent network $\mathcal{G} = (V, \mathcal{E})$ each agent $v_i \in V$ chooses an action $u_i \in U_i$ and communicates with neighbours along the edges $(i, j) \in \mathcal{E}$, receiving a global reward $r(s, u)$. The joint action space $U = \bigtimes U_i$ then grows too large to efficiently control in a centralised fashion [138].

A central distinction is to be made between centralised training with decentralised execution and decentralised training with decentralised execution. In centralised training with decentralised execution, agents can freely communicate during the learning phase, which is performed by a centralised algorithm. The policies are, however, executed in a decentralised fashion, agents communicating only via restricted channels [139]. In decentralised training with decentralised execution, independent learners train policies without explicitly modeling other agents, which provides improved scalability [140].

3) Event-triggered distributed control: A key disadvantage of many distributed control approaches is the necessity of high-frequency communication between the sub-controllers. Solving this issue is the aim of event-triggered control, constructing trigger functions that communicate only when necessary [141]. This can further be extended to a system where the local controllers are deactivated, the control action held constant until a preset condition is met, reducing computational burden [142].

Each local controller continuously monitors its state and transmits updates only if a predefined triggering condition, typically based on state error or performance degradation, is violated [143].

One notable work by Yang et al. [144] combined this with a reinforcement learning approach developing a distributed control scheme tailored to large-scale interconnected nonlinear systems under uncertain couplings. Each local subsystem implements an adaptive-critic architecture to approximate its own optimal control policy directly from collected state-input data. The authors integrate experience replay, buffering past trajectories to accelerate learning and stabilise convergence, and embed event-triggered communication to reduce inter-subsystem data exchange while still guaranteeing closed-loop stability. Crucially, the formulation accommodates unknown subsystem interconnections by treating them as disturbances while enforcing local Lyapunov-based dwell and time-trigger conditions to preserve global stability despite decentralisation. The proposed method is a scalable data-driven algorithm that yields stabilising controllers for each node in the network with bounded communication and minimal modeling assumptions.

Event-triggered data-driven distributed control has found application in a variety of areas. Bu et al. [145] develop a data-driven control algorithm for multimicrogrid interconnected systems with time delays, triggering communication only when the control error exceeds a tunable threshold, proving convergence via Lyapunov theory. Zhu et al. [146] propose a method for connected heterogeneous vehicle platoon control considering sensor faults, network resource usage minimized through the effective use of triggering conditions. Yu et al. [147] develop an iterative learning approach for the control of Multiple High-Speed Trains with switching topologies, proving stability and reduced bandwidth occupancy while retaining sufficient performance.

4) Robustness: The key advantage distributed architectures have over decentralised ones with regard to robustness is the ability to explicitly communicate among subsystems. While this allows for distributed robust control optimisation, it introduces the need for more extensive performance considerations to scale to large-scale systems.

In [148], robustness is achieved by explicitly modeling uncertainty on both subsystem dynamics and interconnection terms, and embedding these uncertainty descriptions into a distributed robust optimisation problem whose constraints are decomposed according to the network graph. Rather than enforcing robustness independently at each node, tightened coupling constraints are coordinated across the network so that worst-case uncertainty propagation is bounded at the system level. This graph-structured decomposition enables robustness guarantees that scale with the degree of interconnection rather than the total number of subsystems, making the approach especially applicable to sparse networks.

A robust distributed model-free controller is constructed in [149] for the domain of volt/VAR regulation over power distribution networks. Each local agent uses robust regression to mitigate bad measurement data and iteratively updates its control strategy via a distributed alternating direction method of multipliers algorithm enhanced with Nesterov acceleration, enabling scalability via complexity growing approximately linearly with the number of direct connections at each subsystem. Efficient robustness is hence achieved via the model-free, data-driven feedback loop which tolerates measurement errors and model mismatch, the distributed optimisation architecture, where only limited peer exchange is required and convergence is accelerated, and decentralised local correction at each node which ensures the system stays robust to both local disturbance and coupling uncertainties.

Overall, robust distributed control for large-scale systems departs from classical robust design by treating uncertainty as a network-level phenomenon rather than simply a collection of independent local disturbances. The reviewed methods demonstrate that scalability depends on the exploitation of interconnection structure, coordinating robustness margins across subsystems.

5) Safety: Recent work in distributed control emphasises certifiable safety through layered protocols that ensure safe interaction among agents under communication and dynamic coupling.

Khaledi et al. [26] constrain a direct data-driven distributed controller for multi-agent systems via CBFs incorporating quadratic programming optimisations. The online computational complexity is dominated by the interior-point quadratic programming problem, which is, for m_i being the control input dimension for agent i, $\mathcal{O}((\sum_i m_i)^3)$ due to the factorization of the Karush-Kuhn-Tucker system for dense data [150]. As a result, this approach is primarily suited to moderately sized networks, highlighting a fundamental limitation of CBF-based safety mechanisms whose safety constraints are enforced through globally coupled online optimisation problems when applied to large-scale systems.

In contrast, the Trajectory-Tube Distributed Explicit Reference Governor proposed in [151] explicitly targets scalability by decoupling safety enforcement across agents through predictive safety tubes that account for higher-order nonlinear dynamics and inter-agent coupling. Safety information is exchanged only locally among neighbouring agents, preventing the propagation of global constraints and enabling safety certificates whose complexity remains bounded as the swarm size increases. This structural localisation of safety guarantees allows the method to scale to large, resource-constrained UAV swarms while maintaining constraint satisfaction under limited communication.

While research on safety in distributed data-driven control architectures remains limited, the presented methods suggest that safety guarantees in large-scale systems are feasible primarily through spatial localisation. Approaches that prevent the global coupling of safety constraints and bound per-agent computation and communication hence seem to be more promising for large-scale system scenarios.

C. Hierarchical control

In hierarchical control scenarios, the control structure is conceptually arranged in multiple levels. The higher level controllers coordinate the actions of the lower level ones, being responsible for meeting the overall goal of the large-scale system [127]. Hierarchical control could be viewed as a special case of distributed control, where some controllers take on special coordination roles.

This paradigm allows coordination of complex large-scale systems by assigning high-level layers to strategic, long-horizon decisions and lower-level layers to fast, local actuation. The approach mitigates the curse of dimensionality inherent in centralised optimisation and can improve robustness and scalability compared to purely distributed schemes [127].

1) Architectural principles: A canonical three-layer hierarchy distinguishes between:

- Supervisory layer (Level 1): Performs system-wide coordination, optimisation, and constraint management. It determines global references or resource allocations for subordinate controllers.
- Coordination layer (Level 2): Translates supervisory commands into feasible sub-objectives for each subsystem, handling coupling constraints and communication among local controllers.
- Local control layer (Level 3): Executes low-level control laws (e.g., MPC, RL, PID) to track references or regulate states under real-time constraints.

This division of control responsibility originates from industrial process control [152] and has since been generalized for power systems [153], transportation networks [90], and multi-robot systems [154]. The separation enables each layer to operate at different sampling rates, leading to computational tractability and improved modularity [128].

2) Data-driven hierarchical model predictive control: Hierarchical Model Predictive Control extends MPC to a multi-level framework, where upper layers solve relaxed or aggregated optimisation problems whose outputs guide lower layers [127].

Shi et al. [155] present an indirect data-driven hierarchical control structure for systems operating under uncertainty. The controller lower in the hierarchy identifies a linear approximation between the higher-level signals and output, whereas the higher-level controller component has the task of dealing with the modeling errors and environment uncertainties.

As for direct data-driven methods, model-free formulations such as hierarchical DeePC have emerged [90], demonstrating such a structure for autonomous mobility-on-demand fleets: the upper layer optimises global vehicle repositioning and the lower layer applies local DeePC to individual agents.

3) Reinforcement Learning: One of the early papers in the field of hierarchical reinforcement learning was [156], proposes a technique of decomposing tasks, solving them individually using Q-modules and sharing solutions across multiple composite tasks, laying the groundwork for the widely successful hierarchical reinforcement learning framework.

Formally, given a high-level controller Q_1 declaring the subgoal g and the lower-level controller Q_2 generating the corresponding action a, hierarchical reinforcement learning can be expressed as follows [157], where Y denotes the Bellman Target:

$$Y_t^{Q^1} = \sum_{t'=t+1}^{t+1+N} R_{t'} + \gamma \max_g (s_{t+1+N}, g) \tag{24}$$

$$Y_t^{Q^2} = R_{t+1} + \gamma \max_a Q(s_{t+1}, a|g) \tag{25}$$

Applying hierarchical reinforcement learning naively to large-scale systems can lead to computational inefficiency and convergence difficulties due to the inherent challenge of a high-dimensional action space, leading to a trend towards combining hierarchical and distributed reinforcement learning, intending to reduce task complexity and accelerate training speeds. One such work is [158], a distributed hierarchical controller using deep reinforcement learning. The authors develop a new framework, coined hierarchical reduction reinforcement learning, addressing the critical challenge of computational inefficiency in deep reinforcement learning for large-scale power grid emergency control. Hierarchical reduction reinforcement learning is based on a two-layer hierarchical decomposition that achieves efficient and accurate action space reduction via a self-supervised learning algorithm. By training a top layer to identify a small effective action space, it significantly reduces the control complexity for the bottom layer. Furthermore, an experiences-sharing-based distributed architecture is integrated to enable parallel training and enhance scalability. This approach substantially improves upon prior art in convergence speed, solution quality, training robustness, and adaptability to large-scale systems.

4) Robustness: Hierarchical control provides a structured mechanism for achieving robustness in complex systems by separating decision-making across layers with distinct time scales and uncertainty characteristics. Rather than enforcing robustness uniformly, hierarchical architectures might confine learning-induced uncertainty and model mismatch to higher layers, while lower layers enforce stability and disturbance rejection, often employing different control methods at each layer. This principle is frequently exploited in robust reinforcement learning architectures, where robustness is achieved by restricting learning to supervisory roles [157, 159].

In the domain of hierarchical control for networked physical systems, the work by Nandakumar et al. [160] demonstrates a data-driven predictive droop-control architecture structured hierarchically applied to islanded microgrids. The high-level module uses a physics-informed sparse identification prediction model and model predictive control to issue set-points for frequency regulation, while the droop control at the lower level tracks those references and compensates disturbances. Reachability analysis is employed to bound worst-case deviations, thereby contributing robustness guarantees under modeling uncertainty and disturbance. Robustness is achieved through adaptive model refinement and time-scale separation, the upper layer continuously re-identifying system dynamics, while the lower predictive droop layer enforces bounded invariance through reachability-based compensation.

Overall, the deliberate separation of uncertainty handling across layers emerges as a key mechanism for achieving robustness in hierarchical control architectures. This structural containment seems particularly critical in large-scale networked systems, where it preserves computational scalability by confining expensive optimisation and adaptation to supervisory layers while allowing local controllers to focus on fast stabilization and disturbance rejection. While existing work on robustness in large-scale hierarchical systems remains limited, the presented results indicate that hierarchical separation offers a promising and scalable pathway for robust data-driven control.

5) Safety: Hierarchical control architectures offer a natural mechanism for enforcing safety in large-scale systems by exploiting abstraction and time-scale separation, thereby distributing safety responsibilities across layers with bounded complexity and dimensionality. Crucially, safety constraints are enforced on low-dimensional abstractions at higher layers and on local state variables at lower layers, avoiding globally coupled safety verification.

A representative example is presented by Ahmad et al. [25], addressing scalability for safe multi-agent reinforcement learning by decoupling coordination from safety enforcement. A high-level policy learns joint cooperative behaviour, while low-level controllers enforce individual safety via CBFs. Safety-critical quadratic programs are solved only at the agent level and depend solely on local state and interaction neighbourhoods, keeping per-agent computational complexity independent of the total number of agents. This approach achieves near-perfect safety rates in dense, conflicting road networks.

In a predictive control setting, Vallon et al. [161] propose a hierarchical, data-driven architecture for unknown operational environments. Offline trajectory data is mined to construct a library of recursively feasible terminal sets and safe equilibria. The supervisory layer selects target states from this library, effectively decomposing the global environment into a sequence of locally reachable, data-validated safe regions. The local MPC problems can then handle the local constraints in parallel, while the high-level hierarchy manages the global topology through a sparse set of historical safe states, providing scalability to a large number of subsystems.

VI. Conclusion

Data-driven control has become a central tool for addressing the scalability and modeling challenges inherent to large-scale dynamical systems. By shifting the emphasis from first-principles modeling to information extracted directly from data, these methods enable control designs that would be infeasible with traditional approaches alone. This survey reviewed the main paradigms in data-driven control, spanning data-driven modeling, direct model-free control, and the architectural choices required to deploy such methods at scale.

A recurring theme across the literature is that scalability is rarely achieved by a single algorithmic idea. Instead, practical large-scale control solutions combine dimensionality reduction, structural decomposition, and appropriate control architectures such as decentralised, distributed, or hierarchical schemes. While data-driven methods can significantly reduce modeling effort and improve flexibility, they introduce new challenges related to computational complexity, stability guarantees and safety certificates.

Despite substantial progress, several open problems remain. These primarily include the exploitation of large-scale structure for robust and safe controllers in networked systems. Future investigation could focus on novel robustness and safety approaches that leverage the overall system topology, instead of relying on the common practice of decomposing global objectives into tendentially conservative local constraints. Further, security aspects, such as malicious agents or subsystems, or the tampering with the learned-from data, are seldom considered. Addressing these challenges will be critical for moving data-driven control toward broadly applicable, reliable solutions for real-world large-scale systems.

References

[1] K. Alzaareer, M. Saad, H. Mehrjerdi, D. Asber, and S. Lefebvre, "Development of new identification method for global group of controls for online coordinated

voltage control in active distribution networks," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 3921–3931, 2020.

[2] Y. Zheng, S. E. Li, K. Li, and W. Ren, "Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed h-infinity controller synthesis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1353–1364, 2017.

[3] P. Kumar, J. B. Rawlings, and S. J. Wright, "Industrial, large-scale model predictive control with structured neural networks," *Computers & chemical engineering*, vol. 150, p. 107 291, 2021.

[4] X. Zhang *et al.*, "Toward scalable multirobot control: Fast policy learning in distributed mpc," *IEEE Transactions on Robotics*, 2025.

[5] W. Tang and P. Daoutidis, "Data-driven control: Overview and perspectives," in *2022 American control conference (ACC)*, IEEE, 2022, pp. 1048–1064.

[6] I. Markovsky, L. Huang, and F. Dörfler, "Data-driven control based on the behavioral approach: From theory to applications in power systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 28–68, 2023.

[7] C. Ma, A. Li, Y. Du, H. Dong, and Y. Yang, "Efficient and scalable reinforcement learning for large-scale network control," *Nature Machine Intelligence*, vol. 6, no. 9, pp. 1006–1020, 2024.

[8] J. Xia, X. Guo, J. H. Park, G. Chen, and X. Xie, "Predictor-based load frequency control for large-scale networked control power systems," *IEEE Transactions on Power Systems*, vol. 39, no. 5, pp. 6263–6276, 2024.

[9] T. Aljohani, M. A. Mohamed, and O. Mohammed, "Tri-level hierarchical coordinated control of large-scale evs charging based on multi-layer optimization framework," *Electric Power Systems Research*, vol. 226, p. 109 923, 2024.

[10] Y. Ma, L. Dai, H. Yang, J. Zhao, R. Gao, and Y. Xia, "Cloud-edge cooperative mpc for large-scale complex systems with input nonlinearity," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 3835–3851, 2024.

[11] P. Verheijen, V. Breschi, and M. Lazar, "Handbook of linear data-driven predictive control: Theory, implementation and design," *Annual Reviews in Control*, vol. 56, p. 100 914, 2023.

[12] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deepc," in *2019 18th European control conference (ECC)*, IEEE, 2019, pp. 307–312.

[13] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[14] 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.

[15] M. Coraggio, D. Salzano, and M. di Bernardo, "Controlling complex systems," in *Reference Module in Materials Science and Materials Engineering*. Elsevier, 2025.

[16] M. Kordestani, A. A. Safavi, and M. Saif, "Recent survey of large-scale systems: Architectures, controller strategies, and industrial applications," *IEEE Systems Journal*, vol. 15, no. 4, pp. 5440–5453, 2021.

[17] N. Yu *et al.*, "Data-driven control, optimization, and decision-making in active power distribution networks," *Applied Energy*, vol. 397, p. 126 253, 2025.

[18] A. C. Antoulas, S. Lefteriu, and A. C. Ionita, "Chapter 8: A tutorial introduction to the loewner framework for model reduction," in *Model Reduction and Approximation*, pp. 335–376. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611974829.ch8.

[19] P. Benner, J.-R. Li, and T. Penzl, "Numerical solution of large-scale lyapunov equations, riccati equations, and linear-quadratic optimal control problems," *Numerical Linear Algebra with Applications*, vol. 15, pp. 755–777, Nov. 2008.

[20] A. K. Prajapati and R. Prasad, "Model reduction using the balanced truncation method and the padé approximation method," *IETE Technical Review*, vol. 39, no. 2, pp. 257–269, 2022.

[21] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.

[22] A. Antoulas, "Approximation of large-scale dynamical systems: An overview," *IFAC Proceedings Volumes*, vol. 37, no. 11, pp. 19–28, 2004, 10th IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Systems 2004: Theory and Applications, Osaka, Japan, 26-28 July, 2004.

[23] R. W. Freund, "Model reduction methods based on krylov subspaces," *Acta Numerica*, vol. 12, pp. 267–319, 2003.

[24] M. Zhang, G. Guo, S. Magnusson, R. C. Pilawa-Podgurski, and Q. Xu, "Data driven decentralized control of inverter based renewable energy sources using safe guaranteed multi-agent deep reinforcement learning," *IEEE Transactions on Sustainable Energy*, vol. 15, no. 2, pp. 1288–1299, 2023.

[25] H. Ahmad *et al.*, *Hierarchical multi-agent reinforcement learning with control barrier functions for safety-critical autonomous systems*, 2025. arXiv: 2507.14850.

[26] M. Khaledi and B. Kiumarsi, "Data-driven distributed safe control design for multi-agent systems," *International Journal of Adaptive Control and Signal Processing*, 2025.

[27] K. P. Wabersich *et al.*, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE*

Control Systems Magazine, vol. 43, no. 5, pp. 137–177, 2023.

[28] Z. Jin, M. Khajenejad, and S. Z. Yong, "Robust data-driven control barrier functions for unknown continuous control affine systems," *IEEE Control Systems Letters*, vol. 7, pp. 1309–1314, 2023.

[29] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[30] C. Wang, Y. Meng, S. L. Smith, and J. Liu, "Data-driven learning of safety-critical control with stochastic control barrier functions," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, IEEE, 2022, pp. 5309–5315.

[31] M. Bajelani and K. van Heusden, "Data-driven input-output control barrier functions," *IEEE Control Systems Letters*, 2025.

[32] J. Borquez, K. Chakraborty, H. Wang, and S. Bansal, "On safety and liveness filtering using hamilton-jacobi reachability analysis," *IEEE Transactions on Robotics*, 2024.

[33] A. Leeman, J. Köhler, S. Bennani, and M. Zeilinger, "Predictive safety filter using system level synthesis," in *Learning for Dynamics and Control Conference*, PMLR, 2023, pp. 1180–1192.

[34] E. Milios, K. P. Wabersich, F. Berkel, and L. Schwenkel, "Stability mechanisms for predictive safety filters," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, IEEE, 2024, pp. 2409–2416.

[35] H. Baumann, A. Schaum, and T. Meurer, "Data-driven control-oriented reduced order modeling for open channel flows," *IFAC-PapersOnLine*, vol. 55, no. 26, pp. 193–199, 2022.

[36] I. V. Gosea, C. Poussot-Vassal, and A. C. Antoulas, "Data-driven modeling and control of large-scale dynamical systems in the loewner framework: Methodology and applications," in *Handbook of Numerical Analysis*, vol. 23, Elsevier, 2022, pp. 499–530.

[37] D. Karachalios, "Data-driven system reduction and identification from input-output time-domain data with the loewner framework," Ph.D. dissertation, Otto-von-Guericke-Universität Magdeburg, 2023.

[38] P. Kergus, S. Formentin, C. Poussot-Vassal, and F. Demourant, "Data-driven control design in the loewner framework: Dealing with stability and noise," in *2018 European Control Conference (ECC)*, IEEE, 2018, pp. 1704–1709.

[39] A. C. Antoulas, I. V. Gosea, and C. Poussot-Vassal, *The loewner framework for parametric systems: Taming the curse of dimensionality*, 2024. arXiv: 2405.00495.

[40] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, "The aaa algorithm for rational approximation," *SIAM Journal on Scientific Computing*, vol. 40, no. 3, A1494–A1522, 2018.

[41] Z. Salehi, P. Karimaghaee, and M.-H. Khooban, "Mixed positive-bounded balanced truncation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 7, pp. 2488–2492, 2021.

[42] I. V. Gosea, S. Gugercin, and C. Beattie, *Data-driven balancing of linear dynamical systems*, 2021. arXiv: 2104.01006.

[43] X. Wang, X. Yang, X. Wang, and Y. Jiang, "Data-driven balanced truncation for second-order systems via the approximate gramians," *Numerical Algorithms*, pp. 1–24, 2025.

[44] P. Benner, P. Kürschner, and J. Saak, "Efficient handling of complex shift parameters in the low-rank cholesky factor adi method," *Numerical Algorithms*, vol. 62, no. 2, pp. 225–251, 2013.

[45] U. Zulfiqar, *Data-driven implementations of various generalizations of balanced truncation*, 2025. arXiv: 2506.23204.

[46] U. Zulfiqar, *Compression and distillation of data quadruplets in non-intrusive reduced-order modeling*, 2025. arXiv: 2501.16683.

[47] S. L. Eskew and J. R. Singler, "A new approach to proper orthogonal decomposition with difference quotients," *Advances in Computational Mathematics*, vol. 49, no. 2, p. 13, 2023.

[48] Q. A. Huhn, M. E. Tano, J. C. Ragusa, and Y. Choi, "Parametric dynamic mode decomposition for reduced order modeling," *Journal of Computational Physics*, vol. 475, p. 111 852, 2023.

[49] S. T. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley, "Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition," *Experiments in Fluids*, vol. 57, no. 3, p. 42, 2016.

[50] T. Nonomura, H. Shibata, and R. Takaki, "Extended-kalman-filter-based dynamic mode decomposition for simultaneous system identification and denoising," *PloS one*, vol. 14, no. 2, e0209836, 2019.

[51] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, "De-biasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets," *Theoretical and Computational Fluid Dynamics*, vol. 31, no. 4, pp. 349–368, 2017.

[52] N. Takeishi, Y. Kawahara, and T. Yairi, "Subspace dynamic mode decomposition for stochastic koopman analysis," *Physical Review E*, vol. 96, no. 3, p. 033 310, 2017.

[53] Y. Ohmichi, Y. Sugioka, and K. Nakakita, "Stable dynamic mode decomposition algorithm for noisy pressure-sensitive-paint measurement data," *AIAA Journal*, vol. 60, no. 3, pp. 1965–1970, 2022.

[54] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz, "Compressed sensing and dynamic mode decomposi-

tion," *Journal of computational dynamics*, vol. 2, no. 2, 2015.
[55] J. N. Kutz, X. Fu, and S. L. Brunton, "Multiresolution dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 2, pp. 713–735, 2016.
[56] N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton, "Randomized dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 4, pp. 1867–1891, 2019.
[57] C. Ocampo-Martinez, S. Bovo, and V. Puig, "Partitioning approach oriented to the decentralised predictive control of large-scale systems," *Journal of Process Control*, vol. 21, no. 5, pp. 775–786, 2011.
[58] L. Bakule and M. Papik, "Decentralized control and communication," *Annual Reviews in Control*, vol. 36, no. 1, pp. 1–10, 2012.
[59] A. Riccardi, L. Laurenti, and B. De Schutter, "A general partitioning strategy for non-centralized control," *arXiv preprint arXiv:2502.21126*, 2025.
[60] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113.
[61] M. A. Basiri and S. Khanmohammadi, *Sindyg: Sparse identification of nonlinear dynamical systems from graph-structured data*, 2025. arXiv: 2409.04463.
[62] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," in *Essays in Econometrics: Collected Papers of Clive W. J. Granger*. USA: Harvard University Press, 2001, pp. 31–47.
[63] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic, "Detecting and quantifying causal associations in large nonlinear time series datasets," *Science Advances*, vol. 5, no. 11, eaau4996, 2019. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.aau4996.
[64] W. Tang and P. Daoutidis, "Network decomposition for distributed control through community detection in input-output bipartite graphs," *Journal of Process Control*, vol. 64, pp. 7–14, 2018.
[65] N. Abbas *et al.*, "Survey of advanced nonlinear control strategies for uavs: Integration of sensors and hybrid techniques," *Sensors*, vol. 24, no. 11, p. 3286, 2024.
[66] M. Abtahi, N. Mojahed, and S. Nazari, "Efficient optimal path planning in dynamic environments using koopman mpc," *arXiv preprint arXiv:2510.02584*, 2025.
[67] W. Zhang, B. Li, H. Xie, and X. Guan, "A deep koopman operators based robust mpc for spacecraft attitude control," *Neurocomputing*, p. 132 313, 2025.
[68] Y. Lin, T. Wen, L. Chen, Q. Wu, and Y. Liu, "Estimating transient stability regions of large-scale power systems part i: Koopman operator and reduced-order model," *CSEE Journal of Power and Energy Systems*, vol. 11, no. 1, pp. 24–37, 2025.
[69] C. Bakker, "Time scale separation and hierarchical control with the koopman operator," *arXiv preprint arXiv:2506.15586*, 2025.
[70] A. Mauroy, Y. Susuki, and I. Mezic, *Koopman operator in systems and control*. Springer, 2020, vol. 7.
[71] L. Shi *et al.*, *Koopman operators in robot learning*, 2024. arXiv: 2408.04200.
[72] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, e0150171, 2016.
[73] G. Meanti, A. Chatalic, V. Kostic, P. Novelli, M. Pontil, and L. Rosasco, "Estimating koopman operators with sketching to provably learn large scale dynamical systems," *Advances in Neural Information Processing Systems*, vol. 36, pp. 77 242–77 276, 2023.
[74] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.
[75] B. A. Belson, J. H. Tu, and C. W. Rowley, "Algorithm 945: Modred - a parallelized model reduction library," *ACM Trans. Math. Softw.*, vol. 40, no. 4, Jul. 2014.
[76] J. H. Tu, "Dynamic mode decomposition: Theory and applications," Ph.D. dissertation, Princeton University, 2013.
[77] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
[78] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 2009.
[79] L. Molgedey and H. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical review letters*, vol. 72, pp. 3634–3637, Jul. 1994.
[80] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2014.
[81] F. Noé and F. Nüske, "A variational approach to modeling slow processes in stochastic dynamical systems," *Multiscale Modeling & Simulation*, vol. 11, no. 2, pp. 635–655, 2013. eprint: https://doi.org/10.1137/110858616.
[82] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, *A kernel-based method for data-driven koopman spectral analysis*, 2015.
[83] C. R. Schwantes and V. S. Pande, "Modeling molecular kinetics with tica and the kernel trick," *Journal of chemical theory and computation*, vol. 11, no. 2, pp. 600–608, 2015.

[84] V. R. Kostic, P. Novelli, A. Maurer, C. Ciliberto, L. Rosasco, and M. Pontil, "Learning dynamical systems via koopman operator regression in reproducing kernel hilbert spaces," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, New Orleans, LA, USA: Curran Associates Inc., 2022.

[85] P. J. Baddoo, B. Herrmann, B. J. McKeon, and S. L. Brunton, "Kernel learning for robust dynamic mode decomposition: Linear and nonlinear disambiguation optimization," *Proceedings of the Royal Society A*, vol. 478, no. 2260, p. 20 210 830, 2022.

[86] P. Liang, J. Feng, J. Su, Q. Guo, and Y. Han, "Fourier kernel-based data driven approximation of the koopman operator," *Nonlinear Dynamics*, pp. 1–13, 2025.

[87] M. Ramadan, E. Toler, and M. Anitescu, *Floodgates up to contain the deepc and limit extrapolation*, 2025. arXiv: 2501.17318.

[88] H. Wang *et al.*, "Mechanical design and data-enabled predictive control of a planar soft robot," *IEEE Robotics and Automation Letters*, 2024.

[89] W. Yu, Z. Tang, and W. Xiong, "Distributed robust data-enabled predictive control based voltage control for networked microgrid system," *Electric Power Systems Research*, vol. 231, p. 110 360, 2024.

[90] P. Zhu, G. Ferrari-Trecate, and N. Geroliminis, "Hierarchical control for vehicle repositioning in autonomous mobility-on-demand systems," *IEEE Transactions on Control Systems Technology*, vol. 33, no. 4, pp. 1463–1476, 2025.

[91] L. Huang, J. Lygeros, and F. Dörfler, "Robust and kernelized data-enabled predictive control for nonlinear systems," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 2, pp. 611–624, 2023.

[92] T. de Jong, S. Weiland, and M. Lazar, *A kernelized operator approach to nonlinear data-enabled predictive control*, 2025. arXiv: 2501.17500.

[93] J. Shi, Y. Lian, and C. N. Jones, "Efficient recursive data-enabled predictive control," in *2024 European Control Conference (ECC)*, IEEE, 2024, pp. 880–887.

[94] J. D. Watson, *Hybrid data-enabled predictive control: Incorporating model knowledge into the deepc*, 2025. arXiv: 2502.12467.

[95] L. Bold, L. Lanza, and K. Worthmann, "Two-component controller design to safeguard data-driven predictive control: A tutorial exemplified with deepc and koopman mpc," *at-Automatisierungstechnik*, vol. 73, no. 6, pp. 383–397, 2025.

[96] S. Li, C. Chen, H. Zheng, J. Wang, Q. Xu, and K. Li, *Robust data-enabled predictive leading cruise control via reachability analysis*, 2024. arXiv: 2402.03897.

[97] E. O. Arwa and K. A. Folly, "Reinforcement learning techniques for optimal power control in grid-connected microgrids: A comprehensive review," *IEEE Access*, vol. 8, pp. 208 992–209 007, 2020.

[98] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[99] L. Busoniu, T. de Bruin, D. Tolic, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annual Reviews in Control*, vol. 46, pp. 8–28, 2018.

[100] P. Barbalho, V. Lacerda, R. Fernandes, and D. Coury, "Deep reinforcement learning-based secondary control for microgrids in islanded mode," *Electric Power Systems Research*, vol. 212, p. 108 315, 2022.

[101] N. Xu, Z. Tang, C. Si, J. Bian, and C. Mu, "A review of smart grid evolution and reinforcement learning: Applications, challenges and future directions," *Energies*, vol. 18, p. 1837, Apr. 2025.

[102] F. B. G. E. Al., "Model-free reinforcement learning in microgrid control: A review," *IEEE Access*, 2025.

[103] T. Sadamoto and A. Chakrabortty, "Fast real-time reinforcement learning for partially-observable large-scale systems," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 3, pp. 206–218, 2021.

[104] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20, Vancouver, BC, Canada: Curran Associates Inc., 2020.

[105] S. Levine, A. Kumar, G. Tucker, and J. Fu, *Offline reinforcement learning: Tutorial, review, and perspectives on open problems*, 2020. arXiv: 2005.01643.

[106] J. Lyu, X. Ma, X. Li, and Z. Lu, "Mildly conservative q-learning for offline reinforcement learning," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, New Orleans, LA, USA: Curran Associates Inc., 2022.

[107] J. Shao, Y. Qu, C. Chen, H. Zhang, and X. Ji, "Counterfactual conservative q learning for offline multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36, Curran Associates, Inc., 2023, pp. 77 290–77 312.

[108] J. Moos, K. Hansel, H. Abdulsamad, S. Stark, D. Clever, and J. Peters, "Robust reinforcement learning: A review of foundations and recent advances," *Machine Learning and Knowledge Extraction*, vol. 4, no. 1, pp. 276–315, 2022.

[109] P. Yu, Z. Wang, H. Zhang, and Y. Song, *Safe reinforcement learning for power system control: A review*, 2024. arXiv: 2407.00681.

[110] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," *Advances in neural information processing systems*, vol. 31, 2018.

[111] T. L. Vu, S. Mukherjee, R. Huang, and Q. Huang, "Barrier function-based safe reinforcement learning for

emergency control of power systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, IEEE, 2021, pp. 3652–3657.
[112] Y. Yang, Y. Jiang, Y. Liu, J. Chen, and S. E. Li, "Model-free safe reinforcement learning through neural barrier certificate," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1295–1302, 2023.
[113] M. Hasanbeig, A. Abate, and D. Kroening, *Cautious reinforcement learning with logical constraints*, 2020. arXiv: 2002.12156.
[114] S. Gu *et al.*, "A review of safe reinforcement learning: Methods, theories and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
[115] S.-H. Wang and E. Davison, "On the stabilization of decentralized control systems," *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 473–478, 1973.
[116] X. Ge, F. Yang, and Q.-L. Han, "Distributed networked control systems: A brief overview," *Information Sciences*, vol. 380, pp. 117–131, 2017.
[117] A. Bemporad and D. Barcelli, "Decentralized model predictive control," *Networked control systems*, pp. 149–178, 2010.
[118] P. Mattsson, F. Bonassi, V. Breschi, and T. B. Schön, "On the equivalence of direct and indirect data-driven predictive control approaches," *IEEE Control Systems Letters*, vol. 8, pp. 796–801, 2024.
[119] J. Berberich and F. Allgower, "An overview of systems-theoretic guarantees in data-driven model predictive control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, 2024.
[120] S. Kofler *et al.*, "Agent-based decentralized model predictive control for plants with multiple identical actuators," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 2, pp. 841–855, 2022.
[121] S. Chen, Z. Wu, and P. D. Christofides, "Decentralized machine-learning-based predictive control of nonlinear processes," *Chemical Engineering Research and Design*, vol. 162, pp. 45–60, 2020.
[122] Z. Karami, Q. Shafiee, Y. Khayat, M. Yaribeygi, T. Dragičević, and H. Bevrani, "Decentralized model predictive control of dc microgrids with constant power load," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 9, no. 1, pp. 451–460, 2019.
[123] X. Shang, J. Wang, and Y. Zheng, "Decentralized robust data-driven predictive control for smoothing mixed traffic flow," *IEEE Transactions on Intelligent Transportation Systems*, 2024.
[124] A. M. Jasim, B. H. Jasim, V. BureÅ¡ And P. MikuleckÃ½, "A new decentralized robust secondary control for smart islanded microgrids," *Sensors*, vol. 22, no. 22, 2022.
[125] Y. Li, H. Zhang, Z. Wang, C. Huang, and H. Yan, "Decentralized control for large-scale systems with actuator faults and external disturbances: A data-driven method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 10 882–10 893, 2023.
[126] M. Hashemnezhad and P. Aristidou, "Safe decentralized voltage regulation in active distribution networks using multi-agent reinforcement learning," *Available at SSRN 5371728*,
[127] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.
[128] J. M. Maestre, R. R. Negenborn, *et al.*, *Distributed model predictive control made easy*. Springer, 2014, vol. 69.
[129] J. Maestre *et al.*, "A comparison of distributed mpc schemes on a hydro-power plant benchmark," *Optimal Control Applications and Methods*, vol. 36, no. 3, pp. 306–332, 2015.
[130] J. Wu, L. Dai, and Y. Xia, "Iterative distributed model predictive control for nonlinear systems with coupled non-convex constraints and costs," *International Journal of Robust and Nonlinear Control*, vol. 34, no. 11, pp. 7220–7244, 2024.
[131] R. Rostami, G. Costantini, and D. Görges, "Admm-based distributed model predictive control: Primal and dual approaches," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 6598–6603.
[132] Z. Wang and C.-J. Ong, "Accelerated distributed mpc of linear discrete-time systems with coupled constraints," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3838–3849, 2018.
[133] C. A. Alonso, J. S. Li, J. Anderson, and N. Matni, "Distributed and localized model-predictive control-part i: Synthesis and implementation," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 2, pp. 1058–1068, 2022.
[134] C. A. Alonso, F. Yang, and N. Matni, "Data-driven distributed and localized model predictive control," *IEEE Open Journal of Control Systems*, vol. 1, pp. 29–40, 2022.
[135] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, "System level synthesis," *Annual Reviews in Control*, vol. 47, pp. 364–393, 2019.
[136] A. Xue and N. Matni, "Data-driven system level synthesis," in *Learning for dynamics and control*, PMLR, 2021, pp. 189–200.
[137] C. A. Alonso, J. S. Li, N. Matni, and J. Anderson, "Distributed and localized model predictive control-part ii: Theoretical guarantees," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 3, pp. 1113–1123, 2023.
[138] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 3, pp. 1086–1095, 2019.

[139] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, “Learning to communicate with deep multi-agent reinforcement learning,” *Advances in neural information processing systems*, vol. 29, 2016.

[140] W. Du and S. Ding, “A survey on multi-agent deep reinforcement learning: From the perspective of challenges and applications,” *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3215–3238, 2021.

[141] Y.-X. Li, Z. Hou, W.-W. Che, and Z.-G. Wu, “Event-based design of finite-time adaptive control of uncertain nonlinear systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3804–3813, 2022.

[142] B. Wang *et al.*, “Event-triggered model predictive control for power converters,” *IEEE transactions on industrial electronics*, vol. 68, no. 1, pp. 715–720, 2020.

[143] S. Hirche *et al.*, “Distributed control for cooperative manipulation with event-triggered communication,” *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1038–1052, 2020.

[144] X. Yang and H. He, “Adaptive critic learning and experience replay for decentralized event-triggered control of nonlinear interconnected systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4043–4055, 2019.

[145] X. Bu, Y. Zhang, X. Ren, and Z. Hou, “Event-triggered data-driven lfc of multi-microgrid interconnected systems with time delays,” *IEEE Transactions on Control of Network Systems*, 2025.

[146] P. Zhu, S. Jin, X. Bu, and Z. Hou, “Distributed data-driven event-triggered fault-tolerant control for a connected heterogeneous vehicle platoon with sensor faults,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 5498–5509, 2024.

[147] W. Yu, D. Huang, Q. Wang, and L. Cai, “Distributed event-triggered iterative learning control for multiple high-speed trains with switching topologies: A data-driven approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10 818–10 829, 2023.

[148] H. Wang, H. Luo, X. Qiao, M. Huo, and X. Xu, “Data-driven distributed robust monitoring and control optimization for interconnected systems,” *IEEE Transactions on Industrial Informatics*, 2024.

[149] S. Li, W. Wu, and Y. Lin, “Robust data-driven and fully distributed volt/var control for active distribution networks with multiple virtual power plants,” *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2627–2638, 2022.

[150] Y. Ye and E. Tse, “An extension of karmarkar’s projective algorithm for convex quadratic programming,” *Mathematical programming*, vol. 44, no. 1, pp. 157–179, 1989.

[151] B. Convens, K. Merckaert, M. M. Nicotra, and B. Vanderborght, “Safe, fast, and efficient distributed receding horizon constrained control of aerial robot swarms,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4173–4180, 2022.

[152] A. J. Terry, *Hierarchical control of production systems*. University of California, Irvine, 1983.

[153] J. C. Vasquez, J. M. Guerrero, J. Miret, M. Castilla, and L. G. De Vicuna, “Hierarchical control of intelligent microgrids,” *IEEE Industrial Electronics Magazine*, vol. 4, no. 4, pp. 23–29, 2010.

[154] C. Ju and H. I. Son, “A hybrid systems-based hierarchical control architecture for heterogeneous field robot teams,” *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1802–1815, 2021.

[155] L. Shi, H. Teng, X. Kan, and K. Karydis, “A data-driven hierarchical control structure for systems with uncertainty,” in *2020 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2020, pp. 57–63.

[156] S. P. Singh, “Transfer of learning by composing solutions of elemental sequential tasks,” *Machine learning*, vol. 8, no. 3, pp. 323–339, 1992.

[157] K. B. Naveed, Z. Qiao, and J. M. Dolan, “Trajectory planning for autonomous vehicles using hierarchical reinforcement learning,” in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 601–606.

[158] Y. Chen, J. Zhu, Y. Liu, L. Zhang, and J. Zhou, “Distributed hierarchical deep reinforcement learning for large-scale grid emergency control,” *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 4446–4458, 2023.

[159] T. Li, K. Srinivasan, M. Q.-H. Meng, W. Yuan, and J. Bohg, “Learning hierarchical control for robust in-hand manipulation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 8855–8862.

[160] A. Nandakumar, Y. Li, Z. Xu, and D. Huang, “Enhancing transient dynamics stabilization in islanded microgrids through adaptive and hierarchical data-driven predictive droop control,” *IEEE Transactions on Smart Grid*, 2024.

[161] C. S. Vallon and F. Borrelli, “Data-driven strategies for hierarchical predictive control in unknown environments,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1434–1445, 2022.