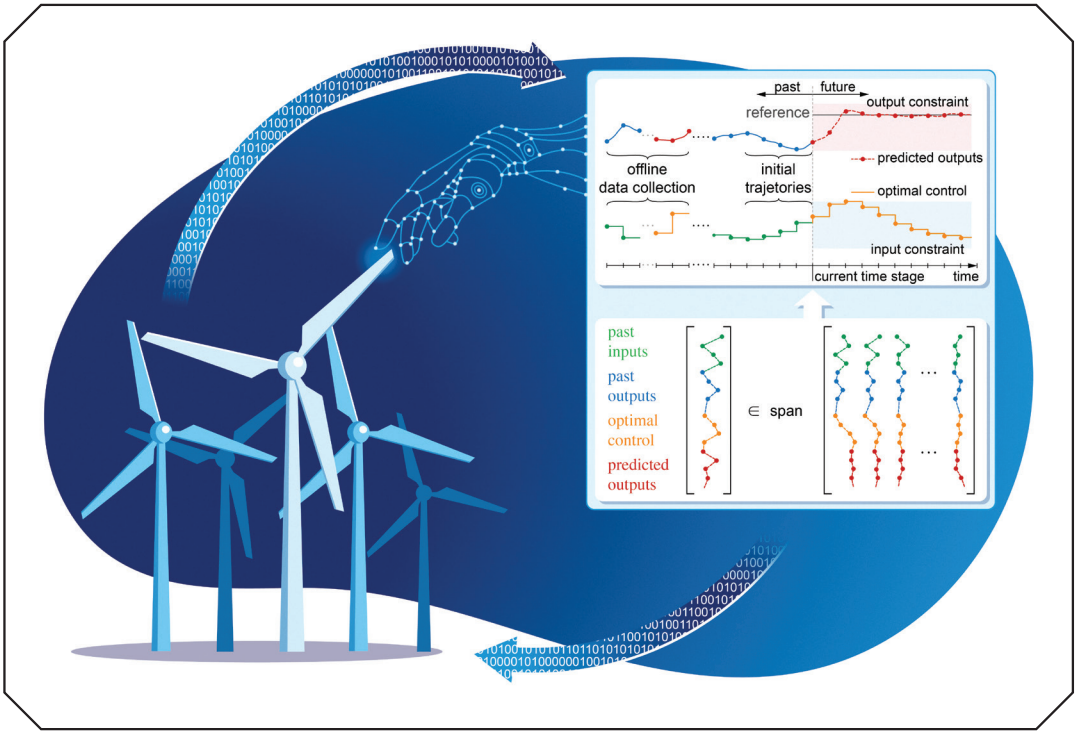
# Data-Driven Control Based on the Behavioral Approach

## FROM THEORY TO APPLICATIONS IN POWER SYSTEMS




SONIA MONTI

IVAN MARKOVSKY, LINBIN HUANG, and FLORIAN DÖRFLER

Physics aims to describe, classify, and predict natural phenomena, while engineering aims to design new or modify existing ones. A phenomenon is characterized by some observed variables. Three common problems control engineers solve are

- *simulation*: predicting the variables in a new experiment
- *smoothing*: removing measurement noise from observations and inferring hidden/latent variables
- *control*: modifying the behavior of some variables by manipulating other variables.

*Digital Object Identifier 10.1109/MCS.2023.3291638*
*Date of current version: 18 September 2023*

To solve them, prior knowledge about the phenomenon is needed. This knowledge is usually given by a model, which is a dynamical system that, ideally, has the same behavior as the real-life phenomenon. The model may be obtained from physical laws (*first principles modeling*), observed data (*black-box modeling*), or a combination of physical laws and observed data (*gray-box modeling*). Modeling using observed data, possibly incorporating some prior knowledge from the physical laws (that is, black-box and gray-box modeling), is called *system identification* (SysID).

SysID is generally applicable and mostly automated. (User input may be needed for tuning hyperparameters.) Modeling from first principles, in contrast, is domain


1066-033X/23©2023IEEE

specific and laborious. Identification methods also allow for an accuracy–complexity tradeoff so that simplified approximate models can be obtained, while modeling from first principles delivers models consistent with data and prior knowledge (unfalsified in the sense of [1]). Thus, SysID is often used for modeling complex phenomena, for which models from first principles are difficult or even impossible to obtain. The approximation aspect of SysID, however, poses an important question: "What is the best approximate model for design?"—that is, "What is the best model for achieving our ultimate goals: simulation, smoothing, and control?" The question gives rise to new areas of research, such as identification for control [2], [3], [4]; dual control [5], [6], [7], [8]; and control-regularized identification [9], [10].

Most design methods are *model-based*—they assume a given model. Recently, an alternative paradigm, called *data-driven*, emerged (see "Summary"). Instead of a model, in the data-driven design paradigm, the prior knowledge about the phenomenon is observed raw data. The aim, then, is to achieve a direct map from the data to the desired solution without identification of a model (see Figure 1).

Since, ultimately, both paths in Figure 1 from data to control are based on data, the somewhat ambiguous term "data-driven" has been used for both. Following [11], we adopt the terminology *indirect* for the path going through a model and *direct* for the path avoiding the model identification step. Still, the separation between direct and indirect is not sharply defined because (as shown later), under certain conditions, the raw data can also be viewed as a model representation. The distinction between direct and indirect methods can be related to the type of model the method is using. Methods using parametric models (such as rational transfer function and state-space representation) are indirect, while methods using nonparametric models (such as the data-driven representation presented in the article) are direct (see "Model-Free Versus Model-Based Methods").

A shift from indirect to direct design is motivated by

- *technological advances*: availability of more data, storage, and computational power
- *system's complexity*: complex physics, environments, or sensing modalities for which first principles models are not available or are not useful for design
- *new methods*: advances in statistical and computational methods for "big data."

From a theoretical point of view, however, the main reason for a shift from indirect to direct design is that the two-step procedure of the indirect design may be suboptimal. Data-driven design is an optimization problem. Given are data, prior knowledge about the data-generating system, and a design criterion. Optimality is with respect to the ultimate goal—the simulated, predicted, or control signal. The indirect model-based approach splits the overall problem into two sequential subproblems: 1) model identification and 2) model-based design; see Figure 2.

The model identification subproblem minimizes a (maximum-likelihood) data-fitting cost function over the model parameters using the data and the prior knowledge about the true data-generating system, but not the design cost function. The model-based design subproblem, on the other hand, minimizes the design cost function using the identified model, but not the data and the prior knowledge about the true data-generating system. This two-step procedure is, in general, suboptimal because, except for

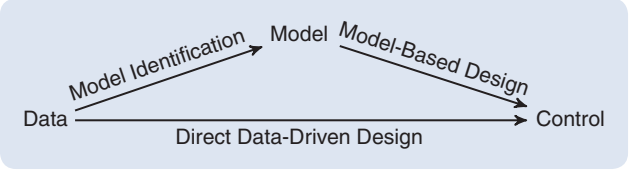



**FIGURE 1** The direct data-driven design paradigm aims to achieve a map from data to result (simulated, smoothed, or control signal) without the identification of a model of the data-generating process.

| | |
|---|---|
| minimize | cost function ($\widehat{w}$) |
| subject to | $\widehat{w}$ is trajectory of model $\mathscr{B}$ |
| where | $\mathscr{B}$ is identified from given data $w_{\mathrm{d}}$ |

**FIGURE 2** Indirect (model-based) data-driven design is a bilevel problem. On the outer level, the cost function, which is a function of a signal $\hat{w}$, is minimized subject to the constraint that $\hat{w}$ is a trajectory of a model $\mathscr{B}$. On the inner level, the model $\mathscr{B}$ is identified from given data $w_{\mathrm{d}}$. The inner problem is also a constrained optimization problem. Its cost function quantifies the fit of $w_{\mathrm{d}}$ by $\mathscr{B}$, and the constraint restricts $\mathscr{B}$ to a given class of models (for example, bounded-complexity linear time-invariant systems with a specified upper bound on the complexity). The indirect data-driven design approach is, thus, modular but generally also suboptimal since there is no general separation principle between these two problems.


## Summary

Behavioral systems theory decouples the notions of a system and representation. A key result is that, under a persistency of excitation condition, the image of a Hankel matrix constructed from the data equals the set of finite-length trajectories of a linear time-invariant system. This result is the cornerstone of a recently emerged approach to direct data-driven control. This self-contained tutorial reviews its foundations and shows how they can be leveraged for data-driven control. We present a generic data-driven interpolation/approximation formulation encompassing many well-known problem instances: among others, finite-horizon data-driven control. We embed this problem formulation into a predictive control setting, robustify it to inexact data by means of regularizations, and apply the resulting methods in the context of power electronics-dominated power systems.

special cases (see [4, Sec. 4]), there is no separation principle for SysID and model-based design. Therefore, an end-to-end direct method may outperform indirect methods.

Additional arguments in favor of the direct approach are the incompatibility of SysID and model-based design methods and the need for model structure selection in SysID. One well-known source of incompatibility of identification and design methods is due to the fact that they deliver or use as an input different parameters and different descriptions of the uncertainty. For example, the identification method may deliver parameters of a Box–Jenkins model with probabilistic and parametric uncertainty bounds, while the design method may require parameters of a state-space model with deterministic and unstructured uncertainty bounds. A conversion of the identified model parameters and their uncertainty estimates can be done in some cases. However, it typically involves conservative approximations and is a nontrivial problem on its own. The structure selection problem refers to restricting the class of candidate models. The model class can be chosen using prior knowledge from physics (that is, gray-box modeling) or using, again, the data. A typical example of structure selection is choosing the model order [12]. The order selection problem is well studied; however, there is still no consensus on a "best" method. Structure selection is a critical step in parametric identification that may introduce a bias error limiting the control performance.

The methods reviewed in this article are based on the behavioral approach to systems theory [13], [107], [108]. The behavioral approach is naturally suited for the direct data-driven design because it views the system as a set of trajectories, thus separating the notion of a system from the one of a representation. In contemporary machine learning language, the behavioral approach is, thus, nonparametric and unsupervised since the data do not have to be labeled into inputs and outputs.

An application of the data-driven methods to a simple but realistic example of a free-falling object in a gravitational field is shown in "Data-Driven Free-Fall Prediction." Free-fall prediction by model-based methods requires knowledge of Newton's second law of motion, the law of gravity, and the law of friction for the modeling of the free-fall phenomenon from first principles. In addition, the model parameters—the gravitational constant and the friction coefficient—have to be known a priori or estimated from data. Prior knowledge of the laws of physics and the physical parameters is not needed for the data-driven prediction, which uses, instead, only observed trajectories of free falls. Moreover, the direct data-driven method is conceptually simple, intuitively clear, and computationally efficient. Finally, the empirical results show that it is more robust to noise than indirect methods using black-box modeling. These insights carry over to more elaborate problems discussed later.

"Classical Versus Behavioral Approach" explains the rationale for the definition of the system as a set, and the "Behavioral Systems Theory" section introduces the notation used in the article. For an in-depth study of the behavioral approach, we recommend a textbook [14], a tutorial article [15], its follow-up [16], and a survey [17] on the behavioral approach from the data-driven perspective.

The "Data-Driven Trajectories Interpolation and Approximation" section presents a generic problem that encompasses simulation, smoothing, and tracking control, among others. The underlying "true system" is implicitly specified by a "data trajectory," and the various problems are solved in this setting as equivalent interpolation/approximation problems. For exact (that is, noise-free) data, we present a direct data-driven solution. For inexact data, we resort to a maximum-likelihood estimator that amounts to a nonconvex optimization problem, and we approach it by either a sequential (indirect) approach or by convex relaxations giving rise to an $\ell_1$-norm regularization.

This problem setup is further refined to the setting of finite-horizon open-loop control in the "Direct Data-Driven Control" section. The data-driven control problem is embedded in a predictive control setting and robustified to inexact data by means of regularizations and estimation constraint softening. The beneficial role of regularization, applications of the method to nonlinear systems, and practical implementation details are discussed, such as hyperparameter tuning, terminal ingredients for closed-loop stability, or efficient real-time computation.

This direct data-driven control method was successfully used across different applications. The "Application to Control of Power Electronics-Dominated Power Systems" section presents different case studies centered around controlling power converters of grid-connected wind turbines. In industrial practice, both the device, grid, and disturbance models are either proprietary or entirely unknown, which motivates the adoption of a data-driven approach. We exemplify different implementation aspects (such as the choice of regularizer and hyperparameter tuning) and showcase advantages of the direct data-driven approach. The code for the examples shown in the tutorial is available at `https://imarkovs.github.io/tutorial`.

## BEHAVIORAL SYSTEMS THEORY

The key insight leading to the behavioral approach—that a dynamical system can be viewed as the set of the trajectories generated by the system—is self-evident. How to use this insight effectively, however, is not evident. This section introduces the notation used in the rest of the article and defines the basic notions of linearity, time invariance, and complexity. Then, we show how the behavioral approach leads to practically useful methods by deriving representations for the finite-horizon

behavior of the system. For discrete-time systems, the representations of the finite-horizon behavior use only basic linear algebra.

### *Linear Time-Invariant Systems*

The concept of a system as a set of trajectories—the behavior—is general. It applies to discrete-event as well as continuous, continuous-time as well as discrete-time, distributed as well as lumped, nonlinear as well as linear, and time-varying as well as time-invariant systems. In this article, the focus is on continuous, discrete-time, lumped, linear, time-invariant systems. A discrete-time, real-valued signal $w$ is a function from the set of natural numbers $\mathbb{N}$ (the *time axis*) to the set of $q$-variate real vectors $\mathbb{R}^q$. The set of all $q$-variate real signals is denoted by $(\mathbb{R}^q)^{\mathbb{N}}$. Often, we restrict the time axis $\mathbb{N}$ to an interval $[1, T]$ of finite length $T$ and denote the signal $w$ restricted to the interval $[1, T]$ by

$$w|_T := (w(1), \ldots, w(T)) \in (\mathbb{R}^q)^T. \tag{1}$$

With some abuse of notation, we consider $w|_T$ as both a finite sequence (1) as well as a vector:

$$w|_T := \begin{bmatrix} w(1) \\ \vdots \\ w(T) \end{bmatrix} \in R^{qT}.$$

## Data-Driven Free-Fall Prediction

Consider a free-falling object in a gravitational field. More specifically, an object is thrown in the air from an initial position $y(0) = y_{\text{ini}}$ with an initial velocity $\dot{y}(0) = v_{\text{ini}}$. The forces acting on the object are gravity (which is proportional to the mass of the object) and friction with the air (which is proportional to the velocity). The goal is to predict the trajectory $y(t)$ for $t > 0$. Since $y$ remains in the plane determined by the initial velocity and the gravity, we can describe the motion in 2D instead of 3D.

Model-based methods require knowledge of the physical parameters—mass, gravitational constant, friction coefficient, and the laws of physics. Data-driven methods use, instead, data collected from repeated experiments of throwing the object and observing its trajectory. The question occurs: "How many experiments should we carry out to collect enough data?" Let $N$ be the number of experiments, and let $\mathscr{Y} = \{y_{\text{d}}^1, \ldots, y_{\text{d}}^N\}$ be the observed data. It turns out that $N = 5$ trajectories are sufficient, provided that their initial conditions $(y_{\text{ini}}^i, v_{\text{ini}}^i)$, $i = 1, \ldots, N$ are linearly independent. The prediction $\hat{y}$ of $y$ is constructed as a linear combination of the observed trajectories

$$\hat{y} := \underbrace{\begin{bmatrix} y_{\text{d}}^1 & \cdots & y_{\text{d}}^N \end{bmatrix}}_{Y_{\text{d}}} \begin{bmatrix} g_1 \\ \vdots \\ g_N \end{bmatrix}. \tag{S1}$$

where $g$ is obtained from the system of linear equations

$$\begin{bmatrix} y_{\text{ini}}^1 & \cdots & y_{\text{ini}}^N \\ v_{\text{ini}}^1 & \cdots & v_{\text{ini}}^N \end{bmatrix} g = \begin{bmatrix} y_{\text{ini}} \\ v_{\text{ini}} \end{bmatrix}. \tag{S2}$$

matching the initial conditions.

The system of equations (S2), however, is underdetermined (four equations and five unknowns). An additional equation is, therefore, needed. The extra equation follows from the prior knowledge that the acceleration $\ddot{y}$ is the same for all trajectories. This gives

$$\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} g = 1. \tag{S3}$$

Next, we show results of a simulation example with free fall in a gravitational field. The initial conditions for the data $\mathscr{Y}$ and the to-be-predicted trajectory $y$ are randomly selected. The continuous-time trajectories are uniformly sampled, which gives us $T_{\text{d}} = 101$-sample discrete-time signals. Zero-mean white uncorrelated Gaussian noise is added to the data $\mathscr{Y}$ to investigate empirically the robustness of the methods to noise.

The compared methods are 1) direct prediction [that is, solve (S2) subject to (S3) and substitute the solution $g$ in (S1) to find the prediction $\hat{y}$] and 2) an indirect method using subspace identification (`n4sid` of the System Identification Toolbox of MATLAB) to identify a model $\hat{\mathscr{B}}$ from the data $\mathscr{Y}$, estimate the initial conditions, and compute the prediction $\hat{y}$ using the model $\hat{\mathscr{B}}$ and the estimated initial conditions.

Figure S1 shows the relative averaged (more than 100 Monte Carlo runs) estimation error $e := 100\% \, \|y - \hat{y}\|_2 / \|y\|_2$, as a function of the noise level. The data $\mathscr{Y}$ consist of $N = 20$ experiments. Both the error and the noise level are given as a percentage of the corresponding true values. The result shows that the direct method is more robust to the noise than the indirect method. A possible explanation for this is that the data-generating system when modeled as an LTI system is neutrally stable (has a pole at one) — a problematic case for standard identification methods.

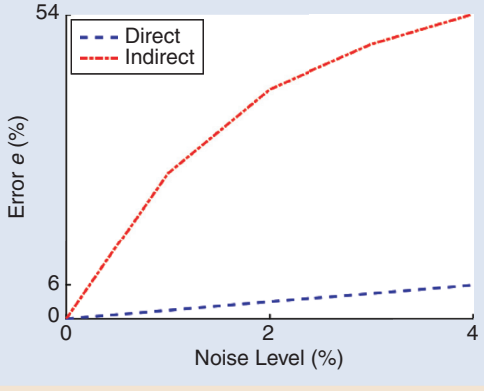



**FIGURE S1** The direct and indirect data-driven simulation of free fall in the presence of noise: the relative average estimation error $e$ of the direct method is uniformly better for all noise levels than the one of the indirect method.

A dynamical system postulates which signals $w$ from the universe of signals $(\mathbb{R}^q)^{\mathbb{N}}$ are possible to be observed. The signals that are possible to observe are called *trajectories* of the system. The set of all trajectories, denoted by $\mathscr{B}$, is called the *behavior* of the system. We identify the system with its behavior so that we use the terms system and behavior interchangeably.

In signal processing and control, we need to specify that

$$\text{“a signal } w \text{ is a trajectory of a system } \mathscr{B}.\text{”} \tag{2}$$

Using the set-theoretic notation of the behavioral approach, this is done by $w \in \mathscr{B}$. What the system $\mathscr{B}$ is (in particular, how it is represented by equations) and how (2) is verified in practice is not detailed at this level of abstraction. The representation and implementation questions are considered at the level of solution methods when different computational procedures for checking (2) are given (see the “Parametric Representations” and “Representation of Restricted Behavior $\mathscr{B}|_T$” sections).

A dynamical system $\mathscr{B}$ is *linear* if $\mathscr{B}$ is a subspace of the space of signals $(\mathbb{R}^q)^{\mathbb{N}}$, that is,

$$w, v \in \mathscr{B} \text{ and } \alpha, \beta \in \mathbb{R} \;\Rightarrow\; \alpha w + \beta v \in \mathscr{B}.$$

The definition makes no assumptions about initial conditions, given input/output partitioning of the variables, and controllability. In particular, the definition applies to autonomous systems.

The system $\mathscr{B}$ is *time invariant* if it is invariant under the action of the shift operator

$$(\sigma w)(t) := w(t+1)$$

that is, $\sigma\mathscr{B} = \mathscr{B}$. Contrary to linearity, there is no simple geometrical interpretation of the time invariance. However, time invariance manifests itself in representations. We denote the set of linear time-invariant (LTI) systems with $q$ variables by $\mathscr{L}^q$.

### Parametric Representations

A subclass of the LTI class, called *finite-dimensional LTI systems*, admits a state-space representation. A state-space

## Classical Versus Behavioral Approach

Traditionally, in systems and control theory, a dynamical system is viewed as a signal processor. A signal processor accepts an input signal and produces an output signal. The rationale for the separation of the signals into input and output is causality: the input causes the output. The input/output relation is formalized mathematically by a function mapping the input to the output, which leads to the ubiquitous notion of a system as an input/output map. The input/output map is visualized as a box with an incoming arrow for the input and an outgoing arrow for the output (see Figure S2).

Many human-made systems, such as filters, software functions, and controllers, operate as signal processors. Some physical phenomena, such as a mass driven by an external force, can be modeled as signal processors, although they do not operate as such (that is, they do not accept an input and produce an output by computation). Moreover, the input/output map is deficient for modeling many human-made systems as well as physical phenomena because it ignores the initial conditions.

Consider a mass driven by an external force. The to-be-modeled variables are the force applied on the mass and the position of the mass. To model the mass as a signal processor, first, the to-be-modeled variables are separated into inputs and outputs. Since the force causes the mass to move, it is natural to choose the force as an input and the position as an output. Note, however, that by applying a force on the mass at some initial moment of time, the position of the mass depends not only on the force, but also on the initial position and velocity, that is, the initial conditions. Thus, modeling the mass driven by an external force as an input/output map ignores the initial conditions (or, equivalently, assumes zero initial conditions).

Formalizing mathematically the effect of the initial conditions led, in the 1960s, to the state-space approach (see Figure S3), which led to major results, such as the Kalman filter, linear quadratic Gaussian control, balanced model reduction, and subspace identification. Its success is due to its generality (it can deal with multivariable linear time-varying systems under nonzero initial conditions) and suitability for numerical computations. Conceptually, the key factor for its success is the shift of perspective from transfer functions to state space.

Most physical phenomena (for example, a mass driven by an external force), however, do not operate as signal

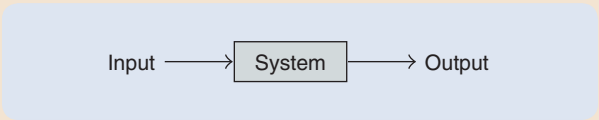



**FIGURE S2** The classical view of a dynamical system is a map from an input signal to an output signal.

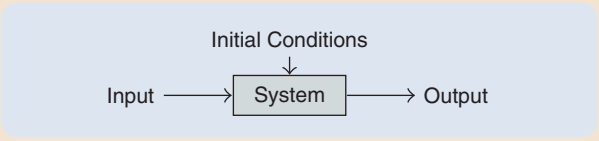



**FIGURE S3** A revision of the input/output map view of the system that takes into account the effect of the initial conditions: The output is a function of the input *and* the initial conditions.

representation is an example of a *parametric representation*, defined by a finite set of parameters. In the classical setting, a parametric representation is *the* system. In the behavioral setting, it is just *a* way of representing the system.

For a permutation $\Pi \in \mathbb{R}^{q\times q}$ and an integer $m$, $0 < m < q$, define a *partitioning* of the variables $w(t) \in \mathbb{R}^q$ into $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^{q-m}$ via

$$\begin{bmatrix} u \\ y \end{bmatrix} := \Pi^{-1} w. \tag{3}$$

Let $\Pi_u$ be the projection of $w$ on the $u$ variable; that is, $\Pi_u w := u$. Acting on a set, $\Pi_u$ projects all elements in the set, which results in a new set. The partitioning (3) is an *input/output partitioning* of $\mathscr{B}$ if $\Pi_u \mathscr{B} = (\mathbb{R}^m)^{\mathbb{N}}$; that is, $u$ is a free variable (which means that $\mathscr{B}$ admits any signal $u \in (\mathbb{R}^m)^{\mathbb{N}}$), the dimension of $u$ is maximal, and the "past" of $y$ does not depend on the "future" of $u$ (causality). The number of inputs $\boldsymbol{m}(\mathscr{B})$ is invariant of the input/output representation and is, therefore, a property of the system $\mathscr{B}$.

An *input/state/output representation* of a finite-dimensional LTI system $\mathscr{B}$ is a vector difference/algebraic equation for an auxiliary variable $x \in (\mathbb{R}^n)^{\mathbb{N}}$, called the *state*, and the output $y \in (\mathbb{R}^p)^{\mathbb{N}}$ that is first order in $x$ and zeroth order in $y$:

$$\begin{bmatrix} \sigma x \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}. \tag{4}$$

The behavior associated with (4) is the solution set of (4) [that is, the trajectories $(u, x, y)$ compatible with (4)] projected on $w$:

$$\mathscr{B}_{\mathrm{ss}}(A, B, C, D, \Pi) := \left\{ w = \Pi \begin{bmatrix} u \\ y \end{bmatrix} \in (\mathbb{R}^q)^{\mathbb{N}} \;\middle|\; \text{there is } x \in (\mathbb{R}^n)^{\mathbb{N}}, \text{ such that (4) holds} \right\}. \tag{5}$$

With some abuse of notation, we refer to $\mathscr{B} = \mathscr{B}_{\mathrm{ss}}(A, B, C, D, \Pi)$ as an input/state/output representation of the system $\mathscr{B}$.

processors. Also, the variables do not come separated into inputs and outputs. Choosing an input/output partitioning (that is, choosing which variables are inputs and which are outputs) becomes a part of the modeling problem. To pose and solve the input/output partitioning problem, a more general setting is needed that does not start with given inputs and outputs.

An a priori fixed input/output partition also leads to issues when interconnecting systems [15]. In the input/output setting, interconnection is an input-to-output assignment—outputs of one system are fed as inputs to another system. For physical phenomena, however, interconnection is equating the variables of one system to the variables of another system. Which variables are equated is determined by physics: inputs may have to be connected to inputs and outputs to outputs. Again, physics clashes with the constraints imposed by the input/output approach.

Confronted with these issues, J. C. Willems created the *behavioral approach* [13], [107], [108]. In the behavioral approach, a dynamical system is defined as a set of trajectories—the *behavior*—without imposing a priori an input/output partitioning of the variables. The change of view from a signal processor to a set of trajectories has far-reaching consequences. It resolves the issues of dealing with nonzero initial conditions, choosing input/output partitioning, and interconnecting systems. Input/output partitioning of the variables is also not needed for defining and studying properties of the system, such as linearity, controllability, and observability. The observer and controller design problems can be defined and solved without postulating inputs and outputs. If necessary, however, all input/output partitions of the variables can be *inferred* from the behavior; that is, the input/output approach is included in the behavioral approach.

Traditionally, the system is identified with a representation, for example, convolution, transfer function, or state space. In the behavioral setting, the system is decoupled from its representations. The behavior is all that matters, while the representations are incidental. Properties of the system and problems involving the system are defined in terms of the behavior. Representations are used to specify the system; to check its properties; and for solving identification, filtering, and control problems that aim at or involve the system. The separation of the system from its representations is at the core of the behavioral approach.

> The operations allowed to bring model equations in a more convenient form are exactly those that do not change the behavior. Dynamic modeling and system identification aim at coming up with a specification of the behavior. Control comes down to restricting the behavior.
>
> —J. C. Willems [15]

The key features distinguishing the behavioral setting from the classical one—defining the system as a set of trajectories and separation of the system from its representations—are two sides of the same coin. Incidentally, these key features of the behavioral approach also make it perfectly suited for the newly emerged data-driven paradigm in systems and control.

The parameters of $\mathscr{B}_{\mathrm{ss}}(A, B, C, D, \Pi)$ are the permutation matrix $\Pi \in \mathbb{R}^{q\times q}$ [which defines the input/output partitioning (3)] and the block matrix

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \mathbb{R}^{(n+p)\times(n+m)}$$

[which defines (4)]. For an autonomous system, (5) becomes

$$\mathscr{B}_{\mathrm{ss}}(A, C) := \{y \in (\mathbb{R}^p)^{\mathbb{N}} \mid \text{there is } x \in (\mathbb{R}^n)^{\mathbb{N}}, \text{ such that } \sigma x = Ax,\ y = Cx\}.$$

The dimension $n$ of the state $x$ is the *order of the representation*. The fact that it is finite is a direct consequence of the finite dimensionality of the system.

The input/state/output representation is called *minimal* if its order is as small as possible over all input/state/output representations of the system. The order $\boldsymbol{n}(\mathscr{B})$ of a minimal input/state/output representation of the system $\mathscr{B}$ is independent of the representation and is called the *order of the system* $\mathscr{B}$.

There are parametric representations of finite-dimensional LTI systems that do not impose an input/output partitioning of the variables. One of them—the kernel representation—is presented in "Kernel Representation and Input/Output Partitions." The kernel representation is a difference equation representation of the system. The degree of the difference equation in a kernel representation is called the *lag* $\ell$ *of the representation* because it equals the number of lags (or delays) in the equation. The minimal lag $\ell(\mathscr{B})$ over all kernel representations of the system is invariant of the representation and is called the *lag of the system*. The lag $\ell(\mathscr{B})$ is equal to the *observability index* of the state-space representation (5), that is, the smallest integer $i$ for which the extended observability matrix with $i$ block rows

$$\mathscr{O}_i(A,C) := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix} \in \mathbb{R}^{pi\times n} \tag{6}$$

achieves a rank equal to $\boldsymbol{n}(\mathscr{B})$. In general, $\ell(\mathscr{B}) \le \boldsymbol{n}(\mathscr{B})$. In the single-output case, $\ell(\mathscr{B}) = \boldsymbol{n}(\mathscr{B})$.

As in the classical setting, the order and the lag of the system are measures of the model's complexity. The number of inputs, however, also determines the complexity. A finite-dimensional system with more inputs is more complex than a system with fewer inputs irrespective of their orders and lags. Therefore, we define *model's complexity* as the ordered triple $\boldsymbol{c}(\mathscr{B}) := (\boldsymbol{m}(\mathscr{B}), \ell(\mathscr{B}), \boldsymbol{n}(\mathscr{B}))$—the number of inputs, lag, and order—and denote with $\mathscr{L}_c^q$ the set of systems with complexity bounded by $c = (m, \ell, n)$, that is, at most $m$ inputs, lag at most $\ell$, and order at most $n$. The rationale for this definition will become clear in the next section when we derive a formula for the dimension of $\mathscr{B}$ restricted to a time interval.

### Representation of Restricted Behavior $\mathscr{B}|_T$

Consider the restriction

$$\mathscr{B}|_T := \{w|_T \mid w \in \mathscr{B}\}$$

of an LTI system $\mathscr{B} \in \mathscr{L}^q$ to the interval $[1, T]$. By the linearity of $\mathscr{B}$, $\mathscr{B}|_T$ is a subspace of $(\mathbb{R}^q)^T$. Its dimension depends on the model's complexity. First, we present an explicit representation of the restricted behavior $\mathscr{B}|_T$ of a finite-dimensional LTI system in terms of the parameters $A, B, C$, and $D$ of a state-space representation. Then, we show alternative nonparametric representations of $\mathscr{B}|_T$ in terms of a trajectory $w_{\mathrm{d}}$ of the system $\mathscr{B}$ that are used in direct data-driven methods [17].

## USING STATE-SPACE REPRESENTATION

Let $\mathscr{B}_{\mathrm{ss}}(A, B, C, D, \Pi)$ be a minimal input/state/output representation of a finite-dimensional LTI system $\mathscr{B}$. For any trajectory $w \in \mathscr{B}|_T$, there is an initial state $x(1) = x_{\mathrm{ini}} \in \mathbb{R}^n$, such that

## Kernel Representation and Input/Output Partitions

A finite-dimensional LTI system $\mathscr{B} \in \mathscr{L}^q$ admits a *kernel representation*

$$\mathscr{B} = \ker R(\sigma) := \{w \in (\mathbb{R}^q)^{\mathbb{N}} \mid R(\sigma)w = 0\} \tag{S4}$$

where the operator $R(\sigma)$ is defined by a $k \times q$ polynomial matrix:

$$R(z) = R_0 + R_1 z + \cdots + R_\ell z^\ell = \begin{bmatrix} R^1(z) \\ \vdots \\ R^k(z) \end{bmatrix} = \begin{bmatrix} R_0^1 + R_1^1 z + \cdots + R_{\ell_1}^1 z^{\ell_1} \\ \vdots \\ R_0^k + R_1^k z + \cdots + R_{\ell_k}^k z^{\ell_k} \end{bmatrix}. \tag{S5}$$

The representation (S4) is called *minimal* if the number of equations $k$ is as small as possible over all kernel representations of $\mathscr{B}$. In a minimal kernel representation, the number of equations $k$ is equal to the *number of outputs* $p$. Also, the *lag* $\ell := \deg R$ is minimal over all kernel representations of $\mathscr{B}$.

Let $\mathscr{B} = \ker R(\sigma)$ be a minimal kernel representation of $\mathscr{B}$. The partitioning (3) is an *input/output partitioning* of $\mathscr{B}$ if and only if $[Q \ -P] := R\Pi$, with $P \in \mathbb{R}^{p\times p}[z]$ nonsingular, where $\mathbb{R}^{p\times p}[z]$ is the set of $p \times p$ matrix polynomials [S1]. The resulting *input/output representation* is

$$\mathscr{B}_{\mathrm{i/o}}(P, Q, \Pi) = \left\{ \Pi \begin{bmatrix} u \\ y \end{bmatrix} \middle| \ Q(\sigma)u = P(\sigma)y \right\}. \tag{S6}$$

#### REFERENCE

[S1] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Trans. Autom. Control*, vol. 36, no. 3, pp. 259–294, Mar. 1991, doi: 10.1109/9.73561.

$$w(t) = \Pi \begin{bmatrix} u(t) \\ y(t) \end{bmatrix}, \quad x(t+1) = Ax(t) + Bu(t),$$
$$y(t) = Cx(t) + Du(t), \quad \text{for } t = 1,2,\ldots,T.$$

This system of equations can be written more compactly as

$$w = \underbrace{\Pi_T \begin{bmatrix} 0_{mT\times n} & I_{mT} \\ \mathscr{O}_T(A,C) & \mathscr{C}_T(H) \end{bmatrix}}_{M_T(A,B,C,D,\Pi)\in\mathbb{R}^{qT\times(n+mT)}} \begin{bmatrix} x_{\text{ini}} \\ u \end{bmatrix}$$

where $\Pi_T \in \mathbb{R}^{qT\times qT}$ is a permutation matrix (determined by $\Pi$), $\mathscr{O}_T(A,C) \in \mathbb{R}^{pT\times n}$ is the extended observability matrix (6), and

$$\mathscr{C}_T(H) := \begin{bmatrix} H(0) & 0 & \cdots & 0 \\ H(1) & H(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ H(T-1) & \cdots & H(1) & H(0) \end{bmatrix} \in \mathbb{R}^{pT\times mT}$$

is the *convolution matrix* with $T$ block rows constructed from the impulse response of the system

$$H(0) = D, \quad H(t) = CA^{t-1}B, \quad \text{for } t = 1,2,\ldots. \tag{7}$$

It follows that

$$\mathscr{B}|_T = \text{image } M_T(A,B,C,D,\Pi). \tag{8}$$

The state-space representation of the restricted behavior (8) is a map from the initial condition $x_{\text{ini}}$ and the input $u$ to a trajectory $w \in \mathscr{B}|_T$. Vice versa, to every $w \in \mathscr{B}|_T$, $T \geq \ell$ correspond a unique $x_{\text{ini}}$ and $u$. The representation (8) shows that, for $T \geq \ell$,

$$\dim \mathscr{B}|_T = \text{rank } M_T(A,B,C,D,\Pi) = mT + n. \tag{9}$$

Here, we used the minimality of the representation and the fact that, for $T \geq \ell$, $\mathscr{O}_T(A,C)$ is a full column rank.

Formula (9) explains why $c = (m, \ell, n)$ is a measure of the model's complexity. Intuitively, the system's complexity is related to its size—the more trajectories $\mathscr{B}$ allows, the more complex it is. For LTI systems, the complexity is then determined by the dimension of $\mathscr{B}$. Formula (9) shows that the dimension of $\mathscr{B}|_T$ is an affine function of $T$ with offset $n$ and slope $m$. For large $T$, meaning $T \geq \ell$, the complexity is, therefore, dominated by the number of inputs $m$. For systems with an equal number of inputs, the more complex system is the one with the larger order.

## USING EXACT RAW DATA

In this section, we consider an exact finite trajectory $w_{\text{d}} \in (\mathbb{R}^q)^{T_{\text{d}}}$ of the system $\mathscr{B}$ as given data. (The subscript "d" stands for "data.") The problem presented is to find conditions under which the data $w_{\text{d}}$ represent the system $\mathscr{B}$ exactly. Later on, we consider the questions of how to design an experiment for collecting data that satisfy the required conditions and methods for dealing with inexact data.

Given a "long" trajectory $w_{\text{d}} \in (\mathbb{R}^q)^{T_{\text{d}}}$ of a system $\mathscr{B} \in \mathscr{L}^q$, multiple "short," $L$-samples-long (with $L < T_{\text{d}}$) trajectories of $\mathscr{B}$ can be created by exploiting the shift-invariance property. A systematic way of doing this is by using the Hankel matrix

$$\mathscr{H}_L(w_{\text{d}}) := \begin{bmatrix} w_{\text{d}}(1) & w_{\text{d}}(2) & \cdots & w_{\text{d}}(T_{\text{d}}-L+1) \\ w_{\text{d}}(2) & w_{\text{d}}(3) & \cdots & w_{\text{d}}(T_{\text{d}}-L+2) \\ \vdots & \vdots & & \vdots \\ w_{\text{d}}(L) & w_{\text{d}}(L+1) & \cdots & w_{\text{d}}(T_{\text{d}}) \end{bmatrix}.$$

The columns of $\mathscr{H}_L(w_{\text{d}})$, viewed as $L$-samples-long signals, are trajectories of $\mathscr{B}|_L$. Combining this fact with (9), it follows that the rank of $\mathscr{H}_L(w_{\text{d}})$ is bounded by $\dim \mathscr{B}|_L = mL + n$.

Consider another $T$-samples-long trajectory $w \in \mathscr{B}|_T$ of $\mathscr{B}$ with $T > \ell$ and $T \leq T_{\text{d}}$. For any $L \in [\ell+1, T]$, we have that

$$\text{rank}\,[\mathscr{H}_L(w_{\text{d}}) \;\; \mathscr{H}_L(w)] \leq mL + n. \tag{10}$$

Inequality (16) expresses the system-theoretic fact that the signal $w$ is a trajectory of the system $\mathscr{B}$ that generated $w_{\text{d}}$ as an algebraic condition—the rank deficiency of a matrix constructed from $w$ and $w_{\text{d}}$. Note that the data-generating system $\mathscr{B}$ is implicit in the Hankel structure and the rank constraint of (10). Therefore, (10) can be used in data-driven signal processing and control for checking the constraint $w \in \mathscr{B}$, where $\mathscr{B} \in \mathscr{L}^q$ is the unknown data-generating system implicitly specified by $w_{\text{d}}$, without involving a representation of the system $\mathscr{B}$.

For any $w_{\text{d}} \in \mathscr{B}|_{T_{\text{d}}}$ and $L \in [1, T_{\text{d}}]$, we have that

$$\text{image } \mathscr{H}_L(w_{\text{d}}) \subseteq \mathscr{B}|_L. \tag{11}$$

We call $w_{\text{d}}$ *generalized persistently exciting or order* $L$ if

$$\text{rank } \mathscr{H}_L(w_{\text{d}}) = mL + n. \tag{12}$$

For $L \geq \ell$, due to (9), the generalized persistency of excitation condition (12) implies that (11) holds with equality; that is,

$$\mathscr{B}|_L = \text{image } \mathscr{H}_L(w_{\text{d}}). \tag{13}$$

Moreover, (12) is a necessary and sufficient condition for (13) [17]. We refer to the representation (13) of the restricted behavior $\mathscr{B}|_L$ as a finite-horizon *data-driven representation* of the system $\mathscr{B}$ because it is expressed directly in terms of the raw data $w_{\text{d}}$.

Under the generalized persistency of excitation condition (12), the rank condition (10) holds with equality. Note that, in the special case $L = T$, the matrix in (10) becomes $[\mathscr{H}_T(w_{\text{d}}) \;\; w]$, which implies that

$$\mathscr{H}_T(w_{\text{d}})g = w \tag{14}$$

for some $g \in \mathbb{R}^{T_d-T+1}$. Since $w$ is an arbitrary element of $\mathscr{B}|_T$, (14) is equivalent to (13). The vector $g$ parameterizes $\mathscr{B}|_T$ in a similar way as the initial condition $x_{ini}$, and the input $u$ parameterizes $\mathscr{B}|_T$ in (8). However, contrary to the parameterization (8), the data-driven parameterization (13) is, in general, redundant—given $w \in \mathscr{B}|_L$, $g$ is not unique.

Choosing the hyperparameter $L$, $\ell+1 \leq L \leq T$ is exploited in the "Data-Driven Trajectories Interpolation and Approximation" section for the development of different methods. The most basic method for solving data-driven problems follows from (14); that is, we choose $L = T$. Then, the constraint $w \in \mathscr{B}|_T$ is equivalent to the existence of a solution $g \in \mathbb{R}^{T_d-T+1}$ of the system of linear equations (14).

The data-driven representation (13) and the generalized persistency of excitation condition (12) can be used in the case of data consisting of multiple trajectories $w_d^1, \ldots, w_d^N$ by replacing the Hankel matrix $\mathscr{H}_L(w_d)$ with the *mosaic Hankel matrix* [18]

$$\mathscr{H}_L(w_d^1, \ldots, w_d^N) := [\mathscr{H}_L(w_d^1) \ \cdots \ \mathscr{H}_L(w_d^N)].$$

Also, (10) has the following generalization for the case when the data consist of multiple trajectories:

$$\operatorname{rank} \mathscr{H}_L(w_d^1, \ldots, w_d^N, w) \leq mL + n.$$

The mosaic Hankel matrix structure is visualized in Figure 3. Other data structures used in data-driven control (presented later) are the page matrix [19] and the trajectory matrix [20]. They are special cases of the mosaic Hankel matrix.

An alternative result to (12) that guarantees the data-driven representation (13) is presented in [21]. It was historically the first result that gives sufficient conditions for a data-driven representation and became known as the *fundamental lemma* in recognition of its importance for SysID and data-driven control [17]. The fundamental lemma was motivated by and has its origin in subspace SysID [22]. An early identifiability condition used in subspace identification involves a state sequence associated with the trajectory $w_d$. The need for general verifiable-from-data identifiability conditions led J. C. Willems and coworkers to the fundamental lemma. Although the result was envisaged as an input design for identifiability, it opened the path to direct data-driven simulation and control [23] as well as to the system-theoretic interpretation of subspace identification methods [24]. An overview of the fundamental lemma and its generalizations is given in "The Fundamental Lemma."

### *Transformations Among Representations*

Methods for checking the properties of the system and solving problems involving the system are based on representations of the system. Different representations lead to different methods with, possibly, different numerical properties. Although by design they compute the same solution, their computational complexity and numerical accuracy may differ. It is important, therefore, to have the flexibility of switching from one representation of the system to another and then use different solution techniques.

Transformations among representations are an essential part of systems theory. The abundance of representations creates many possible paths. Each direct link from one representation to another is a system-theoretic problem. Figure 4 shows the connections among the raw data $w_d$ (which, as we've seen, under certain conditions can be used as a representation of the system), the impulse response, kernel, input/output, and input/state/output representations. The raw data are the least structured (and, therefore, most redundant) way of representing the system. The impulse response representation is valid for any LTI system. However, its description—the impulse response—is, in general, infinite. The impulse response representation is, thus, nonparametric. The kernel, input/output, and input/state/output representations are finitely parameterizable but are limited to finite-dimensional LTI systems. The input/state/output representation is more structured than the kernel representation because it displays, in addition to the finite dimensionality structure, the state and input/output structure of the system.

The transitions from exact raw data to the other representations are exact/deterministic identification problems. The transitions among the representations fall into the

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $w_d^1(1)$ | $w_d^1(2)$ | $w_d^1(3)$ | $w_d^1(4)$ | $w_d^1(5)$ | $w_d^1(6)$ | $w_d^2(1)$ | $w_d^2(2)$ | $w_d^2(3)$ | $w_d^2(4)$ | $w_d^2(5)$ | $w_d^2(6)$ |
| $w_d^1(2)$ | $w_d^1(3)$ | $w_d^1(4)$ | $w_d^1(5)$ | $w_d^1(6)$ | $w_d^1(7)$ | $w_d^2(2)$ | $w_d^2(3)$ | $w_d^2(4)$ | $w_d^2(5)$ | $w_d^2(6)$ | $w_d^2(7)$ |
| $w_d^1(3)$ | $w_d^1(4)$ | $w_d^1(5)$ | $w_d^1(6)$ | $w_d^1(7)$ | $w_d^1(8)$ | $w_d^2(3)$ | $w_d^2(4)$ | $w_d^2(5)$ | $w_d^2(6)$ | $w_d^2(7)$ | $w_d^2(8)$ |
| $w_d^1(4)$ | $w_d^1(5)$ | $w_d^1(6)$ | $w_d^1(7)$ | $w_d^1(8)$ | $w_d^1(9)$ | $w_d^2(4)$ | $w_d^2(5)$ | $w_d^2(6)$ | $w_d^2(7)$ | $w_d^2(8)$ | $w_d^2(9)$ |
| $w_d^1(5)$ | $w_d^1(6)$ | $w_d^1(7)$ | $w_d^1(8)$ | $w_d^1(9)$ | $w_d^1(10)$ | $w_d^2(5)$ | $w_d^2(6)$ | $w_d^2(7)$ | $w_d^2(8)$ | $w_d^2(9)$ | $w_d^2(10)$ |

**FIGURE 3** A mosaic Hankel matrix is a block matrix with Hankel blocks. The color-coded 5 × 12 mosaic Hankel matrix shown has two 5 × 6 scalar Hankel blocks next to each other.

realm of what is called system analysis. In general, going from a less structured representation (for example, impulse response) to a more structured one (for example, state space) makes the problem harder. The problem of transition from the impulse response to an input/state/output representation of the system was solved in the 1960s by R. Kalman and is known as the *realization problem* [28]. The realization problem involves state construction. The state is nonunique and may be nonminimal, which makes the realization problem interesting and challenging. The transition from kernel representation to input/output and input/state/output representations involves choosing an input/output partitioning (see "Kernel Representation and Input/Output Partitions").

### Summary

The behavioral approach detaches the system from its representations. Properties of the system and problems involving the system are defined, then, in terms of the behavior. This leads to new, more general, and more intuitive definitions and problem formulations. Solution methods, however, necessarily use representations of the system. Different representations lead to different methods. The separation of the problem formulation from its solution methods reveals the links among the methods and helps the discovery of new methods. In particular, (13) led to new data-driven signal processing and control methods.

We have presented three characterizations of the constraint that a finite signal $w \in (\mathbb{R}^q)^T$ is a trajectory of an LTI system $\mathscr{B}$. First, using a state-space representation $\mathscr{B}(A, B, C, D, \Pi)$ of $\mathscr{B}$

$$w \in \mathscr{B}|_T \;\Leftrightarrow\; \text{there is } x_{\text{ini}} \in \mathbb{R}^n, \text{ such that } w = M_T(A, B, C, D, \Pi)\begin{bmatrix} x_{\text{ini}} \\ u \end{bmatrix}. \tag{15}$$

Second, using a trajectory $w_{\text{d}} \in \mathscr{B}|_{T_{\text{d}}}$ that satisfies the generalized persistency of excitation condition (12) for an $L \in [\ell + 1, T]$

$$w \in \mathscr{B}|_T \;\Leftrightarrow\; \operatorname{rank}\begin{bmatrix} \mathscr{H}_L(w_{\text{d}}) & \mathscr{H}_L(w) \end{bmatrix} = \operatorname{rank} \mathscr{H}_L(w_{\text{d}}). \tag{16}$$

Third, using a trajectory $w_{\text{d}} \in \mathscr{B}|_{T_{\text{d}}}$ that satisfies the generalized persistency of excitation condition (12) for $L = T$

$$w \in \mathscr{B}|_T \;\Leftrightarrow\; \text{there is } g \in \mathbb{R}^{T_{\text{d}} - T + 1}, \text{ such that } w = \mathscr{H}_T(w_{\text{d}})g. \tag{17}$$

Methods using the state-space representation (15) require SysID, which is typically a nonconvex optimization problem. The direct data-driven methods presented in the following sections use the rank constraint (16) and the system of equations (17). The approach using the rank constraint (16) leads to structured low-rank matrix approximation and completion problems [29], which are nonconvex optimization problems. The approach using the system of equations (17) leads to regularized optimization problems, which are convex [11]. This suggests choosing (17). For $L < T$, however, the rank constraint (16) loosens the generalized persistency of excitation condition. In the extreme $L = \ell + 1$, (16) reduces to parametric (kernel representation) identification of the data-generating system. Thus, the hyperparameter $L$ in (17) controls a transition from a model-based representation to the data-driven representation (17).

## DATA-DRIVEN TRAJECTORIES INTERPOLATION AND APPROXIMATION

This section presents a generic problem that includes simulation, smoothing, and control as special cases.

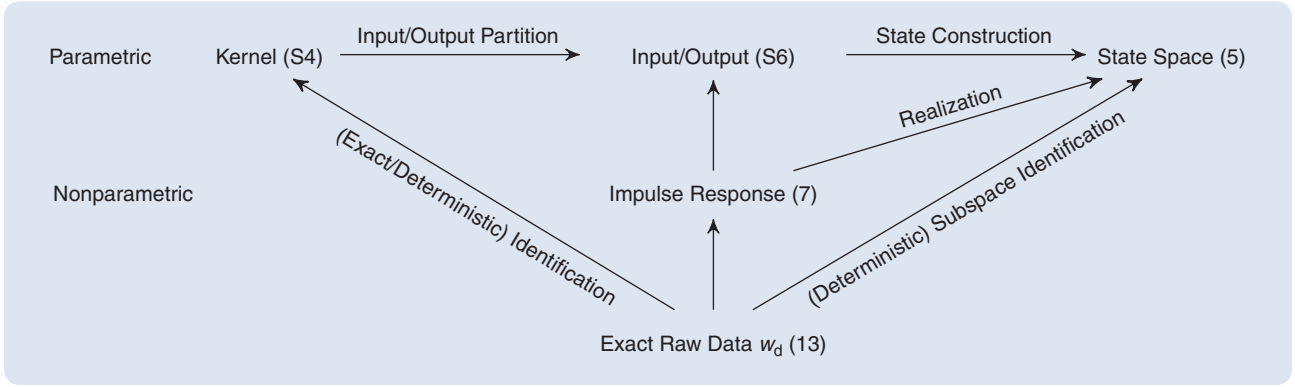



**FIGURE 4** Representations of dynamical systems and some transformations among them. The structure of the system that is implied by the representations grows from bottom to top (raw data, nonparametric, and parametric) and from left to right (no separation of the variables, input/output partition, and state space). Least structured are the raw data, and most structured is the input/state/output representation. The transformations from exact raw data to the other representations are exact/deterministic identification problems. The transition from the impulse response to the state-space representation is the realization problem. The transformations among the parametric representations involve converting the parameter vector of one representation into the parameter vector of another representation.

First, we motivate and formally define the generic problem. Then, we present a direct data-driven solution method based on the data-driven representation (13). The key assumption is that the data trajectory $w_\mathrm{d}$ is exact and satisfies the generalized persistency of excitation condition (12). Finally, we present methods for inexact data $w_\mathrm{d}$, where the inexactness is due to measurement errors, disturbances, or the true data-generating system not being LTI.

### *Problem Formulation*

The interpolation and approximation of functions are basic problems in science and engineering. The solution is sought within a given class of functions. Readers may be familiar with polynomial, piecewise linear, and spline interpolation/approximation. The class of functions that we consider here is trajectories of finite-dimensional LTI systems. For example, in the case of a scalar signal, the problem considered is interpolation via a sum-of-polynomial-times-exponential functions.

The *interpolation* problem aims to recover missing samples of a partially specified trajectory. *Prediction* is a special interpolation problem where "past" samples of a trajectory are given, and "future" samples are missing/to be found. In the interpolation problem, the given data are exact. If this is not the case, for example, due to noise, *approximation* of the data is needed. When the data-generating system is a priori given, the problem is model based, and, when the system is not given (but is implicitly specified by another trajectory), the problem is data driven.

To define the problem, we introduce notation for indicating arbitrary missing elements in a signal $w \in (\mathbb{R}^q)^T$. For a vector $I$ of $K$ indices, $w|_I := [w_{I_1} \cdots w_{I_K}]^\top$ is the subvector of $w \in (\mathbb{R}^q)^T$ with indices $I_1, \ldots, I_K$. Similarly, $\mathscr{H}_T(w_\mathrm{d})|_I$ is the submatrix of $\mathscr{H}_T(w_\mathrm{d})$ with row indices $I_1, \ldots, I_K$. In what follows, $I_\mathrm{given}$ denotes the indices of the *given elements*, and $I_\mathrm{missing}$ denotes the indices of the *missing elements*.

The interpolation/approximation problem considered,

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{w} \quad \|w|_{I_\mathrm{given}} - \hat{w}|_{I_\mathrm{given}}\|_W^2 \\ &\text{subject to} \quad \hat{w} \in \mathscr{B}|_T \end{aligned} \tag{18}$$

is an optimization problem with the decision variable the interpolant/approximant $\hat{w} \in (R^q)^T$ of the given data $w|_{I_\mathrm{given}}$ and the cost function the weighted least-squares approximation error

$$\|w|_{I_\mathrm{given}} - \hat{w}|_{I_\mathrm{given}}\|_W^2 := (w|_{I_\mathrm{given}} - \hat{w}|_{I_\mathrm{given}})^\top W (w|_{I_\mathrm{given}} - \hat{w}|_{I_\mathrm{given}})$$

## The Fundamental Lemma

The problem studied in [21] is to find, for a given finite trajectory $w_\mathrm{d} \in (\mathbb{R}^q)^{T_\mathrm{d}}$ of an LTI system $\mathscr{B}$ and a given natural number $L$, $1 \le L \le T_\mathrm{d}$, conditions under which the "windows" (trajectory snippets) of length $L$

$$\begin{bmatrix} w_\mathrm{d}(1) \\ \vdots \\ w_\mathrm{d}(L) \end{bmatrix}, \begin{bmatrix} w_\mathrm{d}(2) \\ \vdots \\ w_\mathrm{d}(L+1) \end{bmatrix}, \ldots, \begin{bmatrix} w_\mathrm{d}(T_\mathrm{d}-L+1) \\ \vdots \\ w_\mathrm{d}(T_\mathrm{d}) \end{bmatrix} \tag{S7}$$

constructed from the trajectory $w_\mathrm{d}$ span the space $\mathscr{B}|_L$ of all possible trajectories of length $L$ that the system can produce. A compact way of writing that (S7) spans $\mathscr{B}|_L$ is the data-driven representation (13). The problem addressed in [21] is, then, "Under what conditions does (13) hold true?"

The solution given in [21] assumes a given input/output partitioning $w = \begin{bmatrix} u \\ y \end{bmatrix}$ of the variables and provides sufficient conditions only: (13) holds true assuming the following:

- *Assumption S1*: $\mathscr{B}$ is controllable (in the behavioral sense, see [S1, Def. V.1]).
- *Assumption S2*: $u_\mathrm{d}$ is persistently exciting of order $L+n$; that is, $\mathscr{H}_{L+n}(u_\mathrm{d})$ is full row-rank.

Assumption S1 is not verifiable from the data, and assumption S2 requires prior knowledge of the order $n$ of the data-generating system $\mathscr{B}$. The need to assume input/output partitioning and controllability as well as the sufficiency but not necessity of assumptions S1 and S2 make the result of [21] more restrictive than (12). Obtaining conditions for (13) in terms of the input $u_\mathrm{d}$ is motivated from the *input design* perspective [S2], [S3], [S4], [S5]. Assumption S2 can be used for choosing the input so that the data $w_\mathrm{d}$ are guaranteed to ensure (13) for *any* initial condition.

As stated in [21], the extra persistency of excitation of order $n$ in assumption S2 that is needed beyond the obvious persistency of excitation of order $L$ is the crux of the result:

> The interesting, and somewhat surprising, part of Theorem 1 [the fundamental lemma] is that persistency of excitation of order $L+n$ is needed to be able to deduce that the observed sequences (S7) of length $L$ have the "correct" annihilators and the "correct" span. In other words, we have to assume a "deeper" persistency of excitation on $u_\mathrm{d}$ than the width of the windows of $w_\mathrm{d}$ which are considered.

The original publication [21] as well as subsequent ones using and generalizing the result do not give an explanation for this crucial fact. Also, it was not known how conservative assumptions S1 and S2 are. These questions are addressed in [S6], where an alternative constructive proof of the fundamental lemma is given. It shows that assumptions S1 and S2 are nonconservative in the single-input case and characterizes the nongeneric case in which persistency of excitation of order more than the time horizon is needed, relating it to special initial conditions.

defined by a positive-definite weight matrix $W$. When $W$ is diagonal, the problem is *elementwise weighted*. Larger values of weights in an elementwise-weighted problem enforce better approximation of the corresponding data elements. In the limit of weights going to infinity, exact interpolation conditions $w|_{I_{\text{exact}}} = \hat{w}|_{I_{\text{exact}}}$ are approximately enforced for the set of elements $w|_{I_{\text{exact}}}$ corresponding to the large/infinite weights. Dealing with exact interpolation conditions by replacing infinite weights by large finite values, however, is not advised. Apart from involving an ad hoc approximation, it leads to numerical issues due to ill-conditioning of the resulting computational problem. The correct way of dealing with exact interpolation conditions is to solve (18) with the equality constraint $w|_{I_{\text{exact}}} = \hat{w}|_{I_{\text{exact}}}$ [30, Sec. 2].

If an exact interpolant $\hat{w}$ exists, then $w|_{I_{\text{given}}} = \hat{w}|_{I_{\text{given}}}$, and the approximation error $\|w|_{I_{\text{given}}} - \hat{w}|_{I_{\text{given}}}\|_W$ is zero. An exact interpolant may not be unique. Also, equality constraints may cause infeasibility, that is, a lack of solution. The possibilities to recover missing data, approximate inexact data, and interpolate exact data make problem (18) general. "Simulation, Smoothing, and Control as Trajectory Interpolation/Approximation" shows how classical problems fit into the problem formulation (18).

The constraint $\hat{w} \in \mathscr{B}|_T$ in (18) imposes that $\hat{w}$ is a valid trajectory. In the model-based version of the problem, the system $\mathscr{B}$ is a priori given, while, in the data-driven version of the problem, $\mathscr{B}$ is implicitly specified by another given trajectory $w_{\text{d}} \in \mathscr{B}|_{T_{\text{d}}}$. We consider two scenarios for the data trajectory $w_{\text{d}}$:

- $w_{\text{d}}$ is exact; that is, $w_{\text{d}} \in \mathscr{B}|_{T_{\text{d}}}$, where $\mathscr{B} \in \mathscr{L}_{\text{c}}^{q}$.
- $w_{\text{d}}$ is inexact; that is, $w_{\text{d}}$ is not a trajectory of a bounded-complexity LTI system.

With exact data $w_{\text{d}}$ satisfying the generalized persistency of condition (12), the model-based and the data-driven versions of the interpolant/approximant problem (18) are equivalent. Moreover, the problem is convex, so data-driven and model-based methods yield the same solution, despite different numerical properties of the methods.

The equivalence between model-based and data-driven methods does not hold when $w_{\text{d}}$ is inexact. An approximation of the true data-generating system $\mathscr{B}$ by a model $\hat{\mathscr{B}} \in \mathscr{L}_{\text{c}}^{q}$ is needed. It turns out that this modification makes the problem nonconvex. All currently known methods can be viewed as heuristics for solving the underlying nonconvex optimization problem.

### Solution in the Case of Exact Data $w_{\text{d}}$

We start with the data-driven version of the general interpolation/approximation problem (18) assuming exact data

The fundamental lemma was conceived as a theoretical result in SysID; however, due to its applicability in data-driven analysis and control, a number of applications and generalizations appeared. We mention the extensions for

- multiple trajectories [S7] and other matrix structures [25]
- uncontrollable systems [26], [S8]
- other model classes: affine [27], linear parameter varying [S9], linear time varying [S10], flat [S11], delay [S12], finite impulse response Volterra [S13], Wiener–Hammerstein [S14], and bilinear [S15].

#### REFERENCES


[S2] H. van Waarde, "Beyond persistent excitation: Online experiment design for data-driven modeling and control," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 319–324, 2022, doi: 10.1109/LCSYS.2021.3073860.

[S3] A. Iannelli, M. Yin, and R. S. Smith, "Experiment design for impulse response identification with signal matrix models," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 625–630, Sep. 2021, doi: 10.1016/j.ifacol.2021.08.430.

[S4] C. De Persis and P. Tesi, "Designing experiments for data-driven control of nonlinear systems," *IFAC-PapersOnLine*, vol. 54, no. 9, pp. 285–290, Jul. 2021, doi: 10.1016/j.ifacol.2021.06.085.

[S5] J. Berberich, A. Iannelli, A. Padoan, J. Coulson, F. Dörfler, and F. Allgöwer, "A quantitative and constructive proof of Willems' fundamental lemma and its implications," 2022, *arXiv:2208.00905*.

[S6] I. Markovsky, E. Prieto-Araujo, and F. Dörfler, "On the persistency of excitation," *Automatica*, vol 147, Jan. 2023, Art. no. 110657, doi: 10.1016/j.automatica.2022.110657.

[S7] H. 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 Syst. Lett.*, vol. 4, pp. 602–607, Jul. 2020, doi: 10.1109/LCSYS.2020.2986991.

[S8] Y. Yu, S. Talebi, H. van Waarde, U. Topcu, M. Mesbahi, and B. Açikmeş "On controllability and persistency of excitation in data-driven control: Extensions of Willems' fundamental lemma," in *Proc. 60th IEEE Conf. Decis. Contr.*, 2021, pp. 6485–6490, doi: 10.1109/CDC45484.2021.9682952.

[S9] C. Verhoek, R. Tóth, S. Haesaert, and A. Koch, "Fundamental lemma for data-driven analysis of linear parameter-varying systems," in *Proc. 60th IEEE Conf. Decis. Contr.*, 2021, pp. 5040–5046, doi: 10.1109/CDC45484.2021.9683151.

[S10] B. Nortmann and T. Mylvaganam, "Direct data-driven control of linear time-varying systems," 2021, *arXiv:2111.02342*.

[S11] M. Alsalti, J. Berberich, V. G. Lopez, F. Allgöwer, and M. A. Müller, "Data-based system analysis and control of flat nonlinear systems," in *Proc. 60th IEEE Conf. Decis. Contr.*, 2021, pp. 1484–1489, doi: 10.1109/CDC45484.2021.9683327.

[S12] J. G. Rueda-Escobedo, E. Fridman, and J. Schiffer, "Data-driven control for linear discrete-time delay systems," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3321–3336, Jul. 2022, doi: 10.1109/TAC.2021.3096896.

[S13] J. Rueda-Escobedo and J. Schiffer, "Data-driven internal model control of second-order discrete Volterra systems," in *Proc. 59th IEEE Conf. Decis. Contr.*, 2020, pp. 4572–4579, doi: 10.1109/CDC42340.2020.9304122.

[S14] J. Berberich and F. Allgöwer, "A trajectory-based framework for data-driven system analysis and control," in *Proc. Eur. Contr. Conf.*, 2020, pp. 1365–1370, doi: 10.23919/ECC51009.2020.9143608.

[S15] A. Bisoffi, C. De Persis, and P. Tesi, "Data-based stabilization of unknown bilinear systems with guaranteed basin of attraction," *Syst. Contr. Lett.*, vol. 145, Nov. 2020, Art. no. 104788, doi: 10.1016/j.sysconle.2020.104788.

## Model-Free Versus Model-Based Methods

The term *model-free* is sometimes used in the literature as a synonym for what we call, in this article, *direct data-driven*. The qualifier "model-free," however, is ambiguous and may be misleading because it depends on the notion of a model. If a model is understood in the sense of a parametric representation, such as the state-space representations (5), then direct data-driven methods based on the data-driven representation (13) are, indeed, model-free. However, if a model is understood in the sense of a set of trajectories, then direct data-driven methods also use a model. Indeed, the data-driven representation (13) uses image $\mathscr{H}_L(w_d)$ as a model of the data-generating system over the horizon $[1, L]$.

The key question in the model-free versus model-based dilemma is, rather, how the methods impose constraints on the trajectory $\hat{w}$ (see Figure 2). The parametric representations (5) and (S4) restrict $\hat{w}$ by restricting the model's complexity (typically fixing the number of inputs and upper bounding the order or the lag). The data-driven representation (13), in contrast, "lets the data speak" by not restricting a priori the model's complexity. The only constraints that (13) imposes on $\hat{w}$ are the ones already present in the data $w_d$. As shown in the "Models for Inexact Data $w_d$" and "Direct Data-Driven Control" sections, this freedom becomes an issue when the data are not exact because, generically, image $\mathscr{H}_L(w_d) = \mathbb{R}^{qL}$, and, therefore, (13) does not impose constraints. In this case, constraints should be enforced to avoid the trivial model $\hat{\mathscr{B}}|_L = (\mathbb{R}^q)^L$.

One approach of avoiding the trivial model is to impose an upper bound on the model complexity. This leads to parametric model representations and nonconvex optimization problems for the parameters. Another approach is to use unstructured low-rank approximation, which leads to *subspace-type methods*. A third approach is to impose soft constraints by regularization terms in the cost function. The pros and cons of the different approaches are further discussed in the article.

## Simulation, Smoothing, and Control as Trajectory Interpolation/Approximation

The idea that simulation, smoothing, and control can be posed and solved as missing data estimation problems was put forward in [S16]. Here, we show how these problems fit in the interpolation/approximation problem (18) for particular choices of the trajectory $w$. Apart from the input/output partitioning $w = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$, we split the time axis into "past" $w_{\text{ini}}$—the first $T_{\text{ini}}$ samples—and "future" $w_{\text{f}}$—the remaining $T_{\text{f}}$ samples. The past/future partition allows us to take into account initial condition in a representation-free manner (see Figure S4). Thus, in the problems, we specify the four elements of $w$—$u_{\text{ini}}$, $y_{\text{ini}}$, $u_{\text{f}}$, and $y_{\text{f}}$—with their type:

- missing, in which case it is interpolated by $\hat{w}$
- inexact, in which case it is approximated by $\hat{w}$
- exact, in which case it is matched exactly in $\hat{w}$.

A summary of the examples is given in Table S1.

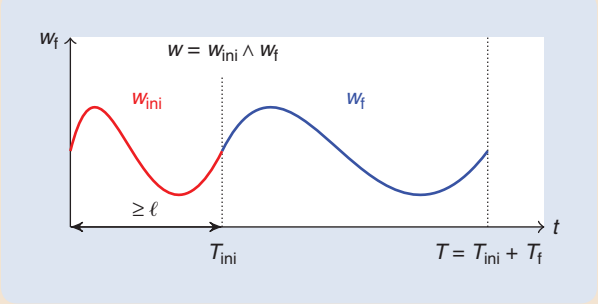



**FIGURE S4** The initial condition for a trajectory $w_{\text{f}} \in \mathscr{B}|_{T_{\text{f}}}$ is specified by a pre-fix trajectory $w_{\text{ini}} \in \mathscr{B}|_{T_{\text{ini}}}$ of length $T_{\text{ini}} \geq \ell$.

### *Simulation*

To define simulation in the behavioral setting, we need to specify the initial condition in a representation-free manner, that is, on the level of trajectories without a state-space representation. This is done by a pre-fix trajectory $w_{\text{ini}}$ to the trajectory $w_{\text{f}}$, for which the initial condition has to be specified. The idea is visualized in Figure S4 and formalized in Lemma S1.

**Lemma S1 (Initial Condition Specification [23])**
Let $\mathscr{B} \in \mathscr{L}^q$ admit an input/output partition $w = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$. Then, for any given $w_{\text{ini}} \in \mathscr{B}_{T_{\text{ini}}}$ with $T_{\text{ini}} \geq \ell$ (where $\ell$ is the lag of $\mathscr{B}$) and $u_{\text{f}} \in (\mathbb{R}^m)^{T_{\text{f}}}$, there is a unique $y_{\text{f}} \in (\mathbb{R}^p)^{T_{\text{f}}}$, such that

$$w_{\text{ini}} \wedge (u_{\text{f}}, y_{\text{f}}) \in \mathscr{B}|_{T_{\text{ini}}+T_{\text{f}}}, \tag{S8}$$

where $w_{\text{ini}} \wedge w_{\text{f}}$ is the concatenation of $w_{\text{ini}}$ and $w_{\text{f}}$.

**TABLE S1** Simulation, Smoothing, and Tracking Control.

| *Problem* | $u_{\text{ini}}$ | $y_{\text{ini}}$ | $u_{\text{f}}$ | $y_{\text{f}}$ |
|---|---|---|---|---|
| Simulation (S9) | Exact | Exact | Exact | Missing |
| Smoothing (S10) | Missing | Missing | Inexact | Inexact |
| Control (S11) | Exact | Exact | Missing | Inexact |

Simulation, smoothing, and tracking control are interpolation/approximation problems (12) for particular choices of the trajectory $w = \left[\begin{smallmatrix} u_{\text{ini}} \\ y_{\text{ini}} \end{smallmatrix}\right] \wedge \left[\begin{smallmatrix} u_{\text{f}} \\ y_{\text{f}} \end{smallmatrix}\right]$, with $u_{\text{ini}}$, $y_{\text{ini}}$, $u_{\text{f}}$, and $y_{\text{f}}$ missing, inexact, or exact.

$w_d \in \mathscr{B}|_{T_d}$. The given data are $w|_{I_{given}}$, that is, the specified elements of the $T$-samples-long trajectories $w$ and the whole $T_d$-samples-long trajectory $w_d$. Consider also a user-defined hyperparameter $L$, which is an integer in the interval $\ell + 1 \leq L \leq T$. Assuming the generalized persistency of excitation condition (12) and using (10) for the constraint of (18), we obtain

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{w} \quad \| w|_{I_{given}} - \hat{w}|_{I_{given}} \|_W^2 \\ &\text{subject to} \quad \operatorname{rank}\left[\mathscr{H}_L(w_d) \;\; \mathscr{H}_L(\hat{w})\right] = mL + n \end{aligned} \tag{19}$$

which is a mosaic Hankel-structured low-rank matrix completion and approximation problem [29]. Due to the rank constraint, (19) is nonconvex. For $L = T$, however, using (14) for the constraint of (18), the problem becomes a linear least-squares problem:

$$\text{minimize} \quad \text{over } g \quad \| w|_{I_{given}} - \mathscr{H}_T(w_d)|_{I_{given}} g \|_W^2. \tag{20}$$

In general, a minimizer $g$ of (20) is not unique. However, any $g$ that minimizes (20) gives rise to the same approximation/interpolation $\hat{w} = \mathscr{H}_T(w_d)g$ of the data $w|_{I_{given}}$. The nonuniqueness of $g$ is effectively used in the case of noisy data by regularization techniques.

In the case of a positive-definite weight matrix $W$, using the least-norm minimizer (computed with the pseudo-inverse †)

$$g = \left(\sqrt{W}\,\mathscr{H}_T(w_d)|_{I_{given}}\right)^{\dagger} \sqrt{W}\, w|_{I_{given}}$$

we obtain a closed-form expression for a solution of (18)

$$\hat{w} = \mathscr{H}_T(w_d)\left(\sqrt{W}\,\mathscr{H}_T(w_d)|_{I_{given}}\right)^{\dagger} \sqrt{W}\, w|_{I_{given}}. \tag{21}$$

If the solution (18) is not unique, the set of all possible solutions is affine and can be characterized explicitly; see [31].

The data-driven interpolation/approximation method (21) has no hyperparameter. Although the model complexity $c = (m, \ell, n)$ is needed for checking the generalized persistency of excitation condition (12), the method does not depend on it. This suggests that the direct data-driven method (21) has an advantage

The simulation problem is defined as follows: given a system $\mathscr{B}$, an input/output partitioning $w = \begin{bmatrix} u \\ y \end{bmatrix}$, a "future" input $u_f \in (\mathbb{R}^m)^{T_f}$, and a pre-fix trajectory $w_{ini} \in (\mathbb{R}^q)^{T_{ini}}$

$$\text{find } y_f \in (\mathbb{R}^p)^{T_f}, \text{ such that (S8) holds.} \tag{S9}$$

The problem in (S9) is an interpolation problem (18) for a partially specified trajectory $w = w_{ini} \wedge (u_f, y_f)$, with $w_{ini}$ and $u_f$ exact and $y_f$ missing.

### Smoothing

The *errors-in-variables Kalman smoothing* problem [S17] is defined as follows: given an LTI system $\mathscr{B}$ and a "noisy" trajectory $w_f = \bar{w}_f + \tilde{w}_f$, where $\bar{w}_{ini} \wedge \bar{w}_f \in \mathscr{B}|_{T_{ini}+T_f}$ for some $\bar{w}_{ini}$ and a zero-mean white Gaussian noise $\tilde{w}_f$ with a covariance matrix that is a multiple of the identity, find $\bar{w}_{ini}$. The maximum-likelihood estimation problem for the initial condition $\bar{w}_{ini}$ is, then

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{w}_{ini} \text{ and } \hat{w}_f \quad \|w_f - \hat{w}_f\|_2^2 \\ &\text{subject to} \quad \hat{w}_{ini} \wedge \hat{w}_f \in \mathscr{B}|_{T_{ini}+T_f}. \end{aligned} \tag{S10}$$

This problem is an interpolation/approximation problem (18), where the initial condition $w_{ini}$ is missing, and the given future trajectory $w_f$ is approximated; see Table S1.

Modifications of the smoothing problem where a disturbance signal is included are possible and fit into the setting of (18).

### Control

In the errors-in-variables Kalman smoothing problem (S10), the data $w_f$ is a "noisy" trajectory of the system $\mathscr{B}$, and the goal is to estimate the initial condition $w_{ini}$. If $w_f$ is a given reference signal (which is not necessarily a trajectory of $\mathscr{B}$), (S10) becomes a *least-squares tracking* problem. In the control setup, the aim is to obtain the signal $\hat{u}_f$, which is the *open-loop* optimal control. Typically, the reference input $u_f$ is zero, so that the minimum energy control is aimed at. The initial condition $w_{ini}$ is unspecified, which means an unknown initial condition.

A modification of the problem where the initial condition is specified and the input is unconstrained is

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{u}_f \text{ and } \hat{y}_f \quad \|y_f - \hat{y}_f\|_2^2 \\ &\text{subject to} \quad w_{ini} \wedge (\hat{u}_f, \hat{y}_f) \in \mathscr{B}|_{T_{ini}+T_f}. \end{aligned} \tag{S11}$$

Problem (S11) is an interpolation/approximation problem (18) with $w_{ini}$ exact, $\hat{u}_f$ missing, and $y_f$ to be approximated; see Table S1.

The least-squares tracking problem is further refined, generalized, and discussed in "Direct Data-Driven Control."

### REFERENCES


[S16] I. Markovsky, "A missing data approach to data-driven filtering and control," *IEEE Trans. Autom. Control*, vol. 62, pp. 1972–1978, Apr. 2017, doi: 10.1109/TAC.2016.2591178.

[S17] I. Markovsky and B. De Moor, "Linear dynamic filtering with noisy input and output," *Automatica*, vol. 41, no. 1, pp. 167–171, Jan. 2005, doi: 10.1016/j.automatica.2004.08.014.

over indirect methods based on parametric model representations for high-complexity systems. This is confirmed empirically in "Data-Driven Simulation of a High-Order System."

### Models for Inexact Data $w_d$

By "$w_\mathrm{d}$ inexact," we mean that there is no bounded-complexity LTI system $\mathscr{B}$ for which $w_\mathrm{d}$ is a trajectory. Prior knowledge about $w_\mathrm{d}$ is needed, then. This prior knowledge is a model. Three commonly used models reflect the source of inexactness—measurement noise, disturbance, and nonlinear dynamics:

- *errors-in-variables*: additive measurement noise on $w_\mathrm{d}$
- *autoregressive moving-average exogenous (ARMAX)*: unobserved disturbances acting on the system and measurement noise on the output $y_\mathrm{d}$
- *nonlinear time-varying system*: the data-generating system is not bounded-complexity LTI.

In the errors-in-variables model, the system $\mathscr{B}$ is bounded-complexity LTI. However, $w_\mathrm{d}$ is observed with measurement noise that is modeled as a stochastic process. In the ARMAX model, the system $\mathscr{B}$ has an additional unobserved input, called *disturbance*, and the output $y_\mathrm{d}$ is observed with measurement noise. The disturbance and the output noise are modeled as stochastic processes. In the stochastic setting of the errors-in-variables and ARMAX models, the goal is to obtain the maximum-likelihood estimator for the true value of the signal $w_\mathrm{d}$. As shown next for the errors-in-variables setup, the maximum-likelihood estimator is defined by a nonconvex optimization problem.

Consider the interpolation/approximation problem (18) in the *errors-in-variables* setup $w_\mathrm{d} = \bar{w}_\mathrm{d} + \tilde{w}_\mathrm{d}$, where $\bar{w}_\mathrm{d} \in \mathscr{B}|_{T_\mathrm{d}}$ is the true value, and $\tilde{w}_\mathrm{d}$ is a zero-mean white Gaussian measurement noise [33]. We assume that the model complexity $c = (m, \ell, n)$ is a priori given. Thus, the prior knowledge is concisely written as $\mathscr{B} \in \mathscr{L}_c^q$. The maximum-likelihood estimation problem for the missing data $w|_{I_\mathrm{missing}}$ with prior knowledge $\mathscr{B} \in \mathscr{L}_c^q$ is

$$\begin{aligned} \text{minimize} \quad & \text{over } \hat{w}, \hat{w}_\mathrm{d}, \hat{\mathscr{B}} \quad \| w|_{I_\mathrm{given}} - \hat{w}|_{I_\mathrm{given}} \|_W^2 \\ & \qquad\qquad + \| w_\mathrm{d} - \hat{w}_\mathrm{d} \|_{W_\mathrm{d}}^2 \\ \text{subject to} \quad & \hat{w} \in \hat{\mathscr{B}}|_T, \hat{w}_\mathrm{d} \in \hat{\mathscr{B}}|_{T_\mathrm{d}}, \text{ and } \hat{\mathscr{B}} \in \mathscr{L}_c^q \end{aligned} \tag{22}$$

where $W$ is the inverse of the covariance matrix of the additive noise on $w|_{I_\mathrm{given}}$, and $W_\mathrm{d}$ is the inverse of the covariance matrix of the measurement noise $\tilde{w}_\mathrm{d}$.

Using (10) with $L = \ell + 1$, (22) becomes a mosaic Hankel-structured low-rank approximation and completion problem:

$$\begin{aligned} \text{minimize} \quad & \text{over } \hat{w}, \hat{w}_\mathrm{d} \quad \| w|_{I_\mathrm{given}} - \hat{w}|_{I_\mathrm{given}} \|_W^2 + \| w_\mathrm{d} - \hat{w}_\mathrm{d} \|_{W_d}^2 \\ \text{subject to} \quad & \operatorname{rank}\left[\mathscr{H}_{\ell+1}(\hat{w}_\mathrm{d})\, \mathscr{H}_{\ell+1}(\hat{w})\right] \le m(\ell+1) + n. \end{aligned}$$

## Data-Driven Simulation of a High-Order System

This sidebar compares the direct method (21) and an indirect method (that is, SysID and model-based simulation) on a step response simulation problem. Any identification method can be used in the indirect approach. In the simulation example, we use the function `n4sid` of the System Identification Toolbox of MATLAB, which implements state-of-the-art subspace identification methods. The data-generating system $\mathscr{B}$ is a random single-input, single-output LTI system of order $n = 100$ (generated in MATLAB with `B = drss(100)`), the data $w_\mathrm{d}$ are a random trajectory of $\mathscr{B}$ with $T_\mathrm{d} = 319$ samples, and $T_\mathrm{f} = 10$ (that is, we aim to compute the first 10 samples of the step response).

To set up (18) for step response simulation, we choose zero initial conditions $w_\mathrm{ini} = 0$ with $T_\mathrm{ini} = n$ and constant unit input $u_\mathrm{f} = 1$. The estimate of the first $T_\mathrm{f}$ samples of the step response of $\mathscr{B}$ is obtained, then, in the block $\hat{y}_\mathrm{f}$ of $\hat{w}$. We verify that the data $w_\mathrm{d}$ satisfy the generalized persistency of excitation condition (12) with $L = n + T_\mathrm{f} = 110$, so that the data-driven representation (13) holds true. To verify that the result computed by (21) is exact, we obtain the first 10 samples $\bar{s}$ of the step response by simulation of $\mathscr{B}$ and compute the error $e = \| \bar{s} - \hat{y}_\mathrm{f} \|$. Its small value $e = 10^{-10}$ is due to the floating point computations and verifies the correctness of the direct method.

In the indirect method, `n4sid` fails with an error message:

```
There are too many parameters to
estimate for chosen estimation data
size. Reduce the model order or use
a larger dataset.
```

Note that a minimal state-space representation $B_\mathrm{ss}(A, B, C, D)$ of $\mathscr{B}$ has $(n+1)^2 = 10201$ parameters. Although this number can be reduced to $2n + 1 = 201$ by using a canonical parameterization, subspace identification methods generally require at least $5n$ data samples. Indeed, the maximal model order accepted by `n4sid` is 59. The computed model $\hat{\mathscr{B}}$ of order 59, however, is not exact. Consequently, the step response $\hat{s}$ computed by the simulation of $\hat{\mathscr{B}}$ has a nonzero error $e = \| \bar{s} - \hat{s} \|$. The error is highly dependent on the data-generating system $\mathscr{B}$.

The example shows that, for high-complexity data-generating systems, the direct approach is more data efficient than indirect approaches due to the high number of parameters to be estimated in the identification step. The data efficiency can be quantified. For data-driven methods based on the representation (13), the minimum number of samples is [17], [32]

$$T_\mathrm{min} := (m+1)L + n - 1$$

where $L$ is the prediction horizon. For indirect methods, the minimum number of samples depends on the identification method but is typically a multiple of $n$, which may exceed $T_\mathrm{min}$.

The problem is nonconvex. Local optimization methods can be used for solving the mosaic Hankel-structured low-rank approximation and completion problem; see [30], [34], and [35]. However, these methods involve a kernel representation of $\hat{\mathscr{B}}$.

Another approach for solving (22) is the *two-step procedure*:

1) Preprocess $w_\mathrm{d}$, aiming to remove the noise.
2) Using the "cleaned" signal $\hat{w}_\mathrm{d}$, find $w|_{I_\mathrm{missing}}$.

The two-step procedure is akin to Figure 2 and reduces the problem with noisy data to the already solved problem with exact data. The signal $\hat{w}_\mathrm{d}$, resulting from step 1, is an estimate of the noise-free signal $\bar{w}_\mathrm{d}$. The maximum-likelihood estimation of $\bar{w}_\mathrm{d}$ from $w_\mathrm{d}$ and the prior knowledge that $\bar{w}_\mathrm{d} \in \mathscr{B}|_{T_\mathrm{d}}$ and $\mathscr{B} \in \mathscr{L}_c^q$ is

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{w}_\mathrm{d},\ \hat{\mathscr{B}} \quad \| w_\mathrm{d} - \hat{w}_\mathrm{d} \|_{W_\mathrm{d}}^2 \\ &\text{subject to} \quad \hat{w}_\mathrm{d} \in \hat{\mathscr{B}}|_{T_\mathrm{d}} \text{ and } \hat{\mathscr{B}} \in \mathscr{L}_c^q. \end{aligned} \tag{23}$$

Problem (23) is a SysID problem. Similarly to (22), (23) can be also be posed and solved as a mosaic Hankel-structured low-rank approximation problem.

Note also that step 2 is equivalent to model-based missing data interpolation/approximation (18) with $\mathscr{B} = \hat{\mathscr{B}}$. Thus, the two-step procedure is a model-based approach for solving (22):

1) *Model identification*: Using $w_\mathrm{d}$, identify a model $\hat{\mathscr{B}}$.
2) *Model-based interpolation/approximation*: Using $\hat{\mathscr{B}}$ and $w|_{I_\mathrm{given}}$, estimate $w|_{I_\mathrm{missing}}$.

In general, the two-step procedure is suboptimal. When $\dim I_\mathrm{given} \leq mT + n$, however, exact interpolation of $w|_{I_\mathrm{given}}$ is possible by any model $\hat{\mathscr{B}}$ with complexity $(m, \ell, n)$. Therefore, $\| w|_{I_\mathrm{given}} - \hat{w}|_{I_\mathrm{given}} \|_W^2 = 0$ and problem (22) decouples into (23) followed by (18) (that is, the two-step procedure is optimal).

A heuristic for preprocessing $w_\mathrm{d}$ is to do rank-$mT + n$ approximation of $\mathscr{H}_T(w_\mathrm{d})$ by truncation of the singular-value decomposition. The heuristic is due to the fact that the Hankel structure is not preserved. The resulting Algorithm 1 is data driven, as it does not derive an LTI model $\hat{\mathscr{B}}$ of $\mathscr{B}$.

Other methods for solving the missing data estimation with noisy data are nuclear norm relaxation [36]

$$\begin{aligned} \text{minimize} \quad &\text{over } \hat{w}_\mathrm{d} \text{ and } \hat{w} \quad \| w|_{I_\mathrm{given}} - \hat{w}|_{I_\mathrm{given}} \|_W^2 \\ &+ \| w_\mathrm{d} - \hat{w}_\mathrm{d} \|_{W_\mathrm{d}}^2 + \gamma \| [\mathscr{H}_T(\hat{w}_\mathrm{d}) \quad \mathscr{H}_T(\hat{w})] \|_* \end{aligned}$$

where the nuclear norm $\| \cdot \|_*$ of a matrix is the sum of its singular values and $\ell_1$-norm regularization of (18)

$$\text{minimize over } g \quad \| w|_{I_\mathrm{given}} - \mathscr{H}_T(w_\mathrm{d})|_{I_\mathrm{given}} g \|_W^2 + \lambda \| g \|_1. \tag{24}$$

The latter is proposed in [11] as a relaxation for solving a related data-driven control problem (see the "Direct Data-Driven Control" section) and is used for data-driven interpolation in [31].

The performance of the methods is compared empirically on data-driven simulation problems using synthetic data corrupted with noise in the errors-in-variables setting and real data from the database for SysID DAISY [37]; see [17, Sec. 4.5] and [31, Sec. 5]. In the "Application to Control of Power Electronics-Dominated Power Systems" section, we validate the performance of data-driven control methods presented in the following section on applications in the energy domain.

### *Summary*

This section demonstrated the synergy between theory and algorithms. The basic direct data-driven interpolation/approximation was presented for LTI systems. The solution (21) was given in closed form. It is hyperparameters free. In the case of noisy data, the maximum-likelihood estimation problem is a mosaic Hankel-structured low-rank approximation, which is nonconvex. Convex relaxation based on $\ell_1$-norm regularization is effective. However, it has a hyperparameter controlling the complexity–accuracy tradeoff that needs ad hoc selection.

## DIRECT DATA-DRIVEN CONTROL

The field of data-driven control has a long and rich history, and it has been gaining significant traction in the last years due to recent advances in technology (the increasing availability of data, storage, and computational power) as well as on the methodological side (particularly, in high-dimensional statistics and optimization algorithms). The field is rapidly expanding, ever more fragmented, and very much in motion at the time of writing. We adopt the classic terminology of adaptive control [38] to provide a classification of different approaches. Namely, we refer to a data-driven control design method as *indirect* if it is based on SysID followed by model-based design. Conversely, it is *direct* when it bypasses models. Another distinction is *certainty equivalence* versus *robust* design, depending on whether uncertainty is taken into account. Often, these classes are not mutually exclusive, and one may interpolate between them [11], [39], [40]; see Figure 5 for a Venn diagram.

In what follows, we focus on methods surrounding the *fundamental lemma* [21] resulting in the data-driven representation (13) of the restricted behavior parameterizing the set of all finite-length trajectories by a matrix data time series. We do not provide comprehensive coverage of the literature but refer readers to a recent survey [17].

**Algorithm 1: Data-Driven Interpolation With Low-Rank Approximation Preprocessing**

**Input:** $w_\mathrm{d}$, $I_\mathrm{given}$, $w|_{I_\mathrm{given}}$, $m$, and $n$.

1: Compute the SVD: $\mathscr{H}_T(w_\mathrm{d}) = U\Sigma V^\top$.
2: Let $r := mT + n$ and let $P \in \mathbb{R}^{qT \times r}$ be the submatrix of $U$ consisting of its $r$ leading columns.
3: Compute $\hat{w} := P(P|_{I_\mathrm{given}})^\dagger w|_{I_\mathrm{given}}$.

**Output:** $\hat{w}$.

In particular, we do not cover explicit control policies derived from the fundamental lemma; compare [41], [42], and [43]. Rather, we focus on an implicit receding-horizon-control approach, as in classic *model predictive control* (*MPC*) augmented with *moving horizon estimation* (*MHE*) [44], [45].

At first glance, it seems unconventional to bring the elegant, scholarly, and abstract concept of behavioral systems theory together with MPC, which is a pragmatic, computational, and highly applicable method. Indeed, Willems [46] was ambivalent:

> For my own taste, it [MPC] has perhaps too little system theory and too much brute force computation in it, but MPC is an area where essentially all aspects of the field, from modeling to optimal control, and from observers to identification and adaptation, are in synergy with computer control and numerical mathematics.

For context, we remark that this quote dates back to 2007, when the system theory of MPC and MHE was much less developed as compared to today. Aside from the evergreen discussion of scientific elegance versus brute-force computation, we should observe that the core concept of behavioral systems theory, the set of trajectories, is also the main side constraint of MPC and MHE. In a data-driven context, what makes behavioral systems theory useful is the very fact that it is abstract, that is, representation free: the set of trajectories can be spanned by a parametric model or mere data, as in (13).

### Open-Loop Linear Quadratic Tracking: From Indirect to Direct Data-Driven Formulations

We start our journey with indirect and certainty-equivalent data-driven control design and then move toward direct and robust approaches. In particular, we begin with a refined version of the data-driven approximation problem (18). Consider the open-loop linear quadratic (LQ) optimal tracking control problem

$$\begin{aligned}&\underset{u_{\mathrm{f}},y_{\mathrm{f}}}{\text{minimize}} \sum_{t=1}^{T_{\mathrm{f}}} \|y_{\mathrm{f}}(t)-y_{\mathrm{r}}(t)\|_Q^2+\|u_{\mathrm{f}}(t)-u_{\mathrm{r}}(t)\|_R^2\\ &\text{subject to } (u_{\mathrm{ini}},y_{\mathrm{ini}})\wedge(u_{\mathrm{f}},y_{\mathrm{f}})\in\mathscr{B}|_{T_{\mathrm{ini}}+T_{\mathrm{f}}}\end{aligned} \tag{25}$$

on a finite horizon $T_{\mathrm{f}}>0$, where $w_{\mathrm{r}}=(u_{\mathrm{r}},y_{\mathrm{r}})\in\mathbb{R}^{qT_{\mathrm{f}}}$ is a user-defined reference trajectory (not necessarily in $\mathscr{B}|_{T_{\mathrm{f}}}$), and $w_{\mathrm{f}}=(u_{\mathrm{f}},y_{\mathrm{f}})\in\mathbb{R}^{qT_{\mathrm{f}}}$ is the future trajectory of length $T_{\mathrm{f}}\geq 1$ to be designed. Further, $Q\succeq 0$ and $R\succ 0$ are user-defined weighting matrices, where $\succ$ ($\succeq$) and $\prec$ ($\preceq$) denote positive and negative (semi)definiteness, respectively, and $\|e\|_Q=\sqrt{e^TQe}$ is a (semi-)norm for $Q\succeq 0$. Finally, $w_{\mathrm{ini}}=(u_{\mathrm{ini}},y_{\mathrm{ini}})$ is a given pre-fix trajectory of length $T_{\mathrm{ini}}\geq\ell$ setting the initial condition $x_{\mathrm{ini}}$ akin to MHE [23]. Indeed, when resorting to a state-space representation, $x_{\mathrm{ini}}$ can be uniquely recovered from

$$y_{\mathrm{ini}}=\mathscr{O}_{T_{\mathrm{ini}}}(A,C)x_{\mathrm{ini}}+\mathscr{C}_{T_{\mathrm{ini}}}(A,B,C,D)u_{\mathrm{ini}}$$

since $\mathscr{O}_{T_{\mathrm{ini}}}(A,C)$ has full rank for $T_{\mathrm{ini}}\geq\ell$; see also Lemma S1.

Predictive control problems, such as (25), are often augmented with terminal ingredients to ensure closed-loop stability for receding-horizon implementation. This topic will be addressed in the later sections, and we focus on the open loop for now.

The LQ control problem (25) can be solved by a variety of methods provided that a model of $\mathscr{B}$ is available [47], typically a state-space model as in (8). In this case, (25) becomes

$$\begin{aligned}&\text{minimize} \quad \text{over } u_{\mathrm{f}},\ y_{\mathrm{f}},\ x_{\mathrm{ini}} \quad \|y_{\mathrm{f}}-y_{\mathrm{r}}\|_Q^2+\|u_{\mathrm{f}}-u_{\mathrm{r}}\|_R^2\\ &\text{subject to} \quad \begin{bmatrix}y_{\mathrm{ini}}\\ y_{\mathrm{f}}\end{bmatrix}=\mathscr{O}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(A,C)x_{\mathrm{ini}}+\mathscr{C}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(A,B,C,D)\begin{bmatrix}u_{\mathrm{ini}}\\ u_{\mathrm{f}}\end{bmatrix}\end{aligned} \tag{26}$$

where (with a slight abuse of notation) we redefined $Q$ and $R$ as $\mathrm{blkdiag}(Q,\ldots,Q)$ and $\mathrm{blkdiag}(R,\ldots,R)$, respectively. The formulation (26) is standard in MPC and MHE. We now focus on the LQ problem if no model and only raw data are available.

In a conventional indirect setting, SysID and LQ control are sequential: find the best control subject to a model, where the model itself is fitted to data. To formalize this idea, consider data $w_{\mathrm{d}}=(u_{\mathrm{d}},y_{\mathrm{d}})$ collected offline and assembled in a Hankel matrix $\mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(w_{\mathrm{d}})$. According to $w_{\mathrm{ini}}=(u_{\mathrm{ini}},y_{\mathrm{ini}})$ and $w_{\mathrm{f}}=(u_{\mathrm{f}},y_{\mathrm{f}})$, permute and partition the Hankel matrix as

$$\begin{bmatrix}w_{\mathrm{ini}}\\ w_{\mathrm{f}}\end{bmatrix}\sim\begin{bmatrix}u_{\mathrm{ini}}\\ u_{\mathrm{f}}\\ y_{\mathrm{ini}}\\ y_{\mathrm{f}}\end{bmatrix},\ \mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(w_{\mathrm{d}})\sim\begin{bmatrix}U_{\mathrm{p}}\\ U_{\mathrm{f}}\\ Y_{\mathrm{p}}\\ Y_{\mathrm{f}}\end{bmatrix}=\begin{bmatrix}\mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(u_{\mathrm{d}})\\ \mathscr{H}_{T_{\mathrm{ini}}+T}(y_{\mathrm{d}})\end{bmatrix}$$

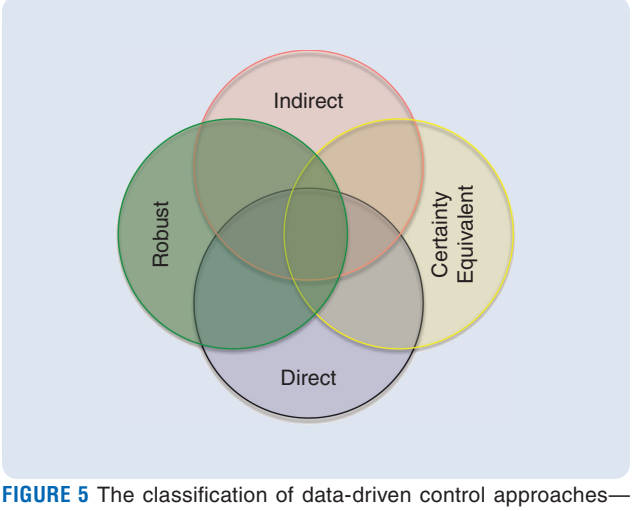



**FIGURE 5** The classification of data-driven control approaches—direct, indirect, robust, and certainty equivalent—displayed as Venn diagram.

where $\sim$ denotes similarity under a permutation of the rows. As an identifiability condition, we assume that the data $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d})$ are persistently exciting; that is, $\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})$ is of rank $m \cdot (T_\mathrm{ini} + T_\mathrm{f}) + n$, where $T_\mathrm{ini} \geq \ell$ and $(\ell, n)$ are a priori estimates (or upper bounds) on the lag and order of the underlying system.

In the indirect setting, we seek a linear model from data. In particular, having in mind the finite-horizon LQ optimal control problem (25), we seek a linear multi-step predictor, that is, a matrix $K$ relating past as well as future inputs and outputs:

$$y_\mathrm{f} = \underbrace{\begin{bmatrix} K_\mathrm{p} & K_\mathrm{f} \end{bmatrix}}_{=K} \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u_\mathrm{f} \end{bmatrix}. \tag{27}$$

Alternatively, of course, a single-step predictor (for example, in the form of a state-space model) can be identified, and a multistep predictor can be built recursively as in (26). We refer to [48] for a general discussion on multistep versus single-step predictors.

The multistep predictor $K$ can also be found from data by replacing the variables $(u_\mathrm{ini}, y_\mathrm{ini}, u_\mathrm{f}, y_\mathrm{f})$ in (27) by Hankel matrix data $(U_\mathrm{p}, Y_\mathrm{p}, U_\mathrm{f}, Y_\mathrm{f})$ and solving for $K$ in a least-squares sense

$$K = \operatorname*{argmin}_{\hat{K}} \left\| Y_\mathrm{f} - \hat{K} \cdot \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} \right\|_F = Y_\mathrm{f} \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix}^\dagger \tag{28}$$

where $\| \cdot \|_F$ denotes the Frobenius norm. The multistep predictor (27)–(28) is well known in the context of *subspace predictive control* (*SPC*) [49], [50]; see "Subspace Predictive Control." It is educational to compare this purely data-based multistep predictor to the model-based predictor in (26) parameterized by $A$, $B$, $C$, and $D$. For exact data, we have that $\operatorname{rank}(K_\mathrm{p}) = n$ (ensuring LTI behavior of desired complexity) and a lower block-triangular zero pattern of $K_\mathrm{f}$ (ensuring causality).

Thus, the indirect approach [least-squares ID (28) followed by LQ control (25)] can be formalized as a *bilevel* program

$$\begin{aligned} \text{minimize} \quad & \text{over } u_\mathrm{f}, y_\mathrm{f} \quad \| y_\mathrm{f} - y_\mathrm{r} \|_Q^2 + \| u_\mathrm{f} - u_\mathrm{r} \|_R^2 \\ \text{subject to} \quad & y_\mathrm{f} = K \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u_\mathrm{f} \end{bmatrix} \\ & K = \operatorname{argmin} \text{over } \hat{K} \left\| Y_\mathrm{f} - \hat{K} \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} \right\|_F. \end{aligned} \tag{29}$$

We now present a direct data-driven approach to LQ control. The fundamental lemma implies that the concatenated initial and future trajectory $w := w_\mathrm{ini} \wedge w_\mathrm{f} \in \mathscr{B}|_{T_\mathrm{ini}+T_\mathrm{f}}$ lies in the image of $\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})$. Thus, $w = \mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d}) g$ for some $g$, and the LQ control problem (25) can be posed as

$$\begin{aligned} \text{minimize} \quad & \text{over } u_\mathrm{f}, y_\mathrm{f}, g \quad \| y_\mathrm{f} - y_\mathrm{r} \|_Q^2 + \| u_\mathrm{f} - u_\mathrm{r} \|_R^2 \\ \text{subject to} \quad & \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{f} \end{bmatrix} g = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u_\mathrm{f} \\ y_\mathrm{f} \end{bmatrix}. \end{aligned} \tag{30}$$

This direct LQ control formulation has been proposed in [23]. Its interpretation is intuitive from the perspective of dictionary learning. The columns of the Hankel matrix $\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})$ serve as a trajectory library spanning $\mathscr{B}|_{T_\mathrm{ini}+T_\mathrm{f}}$. These are, then, linearly combined through the vector $g$ to match the pre-fix trajectory $w_\mathrm{ini} = (u_\mathrm{ini}, y_\mathrm{ini})$ and to optimally synthesize the future control trajectory $w_\mathrm{f} = (u_\mathrm{f}, y_\mathrm{f})$; see Figure 6 for a schematic illustration. The general idea of synthesizing optimal trajectories by linearly combining elements from a library appears throughout the field and in applications, for example, in the context of motion primitives in robotics for motion planning [51] and predictive control [52].

Of course, all formulations (25)–(30) are identical for exact data. For inexact data, the relations among these formulations have been studied in [11], [17], [54], [55], and [56], and we will resume this discussion when it comes to suitable regularizations.

## Subspace Predictive Control

By substituting the explicit least-squares solution (28) into (29) one arrives at the subspace predictive control (SPC) problem formulation coined in [49]

$$\begin{aligned} \text{minimize} \quad & \text{over } u_\mathrm{f}, y_\mathrm{f} \quad \| y_\mathrm{f} - y_\mathrm{r} \|_Q^2 + \| u_\mathrm{f} - u_\mathrm{r} \|_R^2 \\ \text{subject to} \quad & y_\mathrm{f} = Y_\mathrm{f} \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix}^\dagger \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u_\mathrm{f} \end{bmatrix}. \end{aligned} \tag{S12}$$

In the case of inexact data, SPC is often augmented with further pre- and postprocessing of the data, for example, low-rank approximation (typically, via singular-value thresholding) to select the model complexity and thresholding of matrix elements to arrive at a causal (lower-triangular) sparsity. SPC has a long and rich history, is easy to implement, and has demonstrated excellent performance in numerous case studies; see [50] for a review.

In relation to DeePC (31), the main difference is that SPC is focused on the predictor (S12), whereas DeePC (31) is focused on the prediction itself. Hence, the denoising and robustification schemes are different (for example, low-rank approximation versus regularization). However, the two methods become identical when using the projection-based regularizer (35) with sufficiently large coefficient $\lambda_g$ [11].

A comparison of the direct and indirect problem formulations reveals the usual pros and cons, which have often been elaborated upon [4]. For example, the indirect approach consists of modular and well-understood steps. However, it requires model selection and pre-/postprocessing of the data, it is hard to propagate uncertainty, and it is often regarded as an "art" and cumbersome by practitioners. Conversely, the last direct formulation (30) is computationally more involved (more decision variables and less structured matrices), which reveals a strength of parametric models: they are compact, cleaned up, and denoised representations. However, the direct approach is end to end; is arguably simple; and holds the promise to extend beyond deterministic LTI systems, to be adaptive (that is, the data matrices could be updated online), and to improve over the sequential bilevel approach (29) (since there is no separation principle for the nested optimization problems). It is the starting point for the expanding literature on data-enabled predictive control (DeePC) presented next, and we will see how to robustify it in the case of noisy data or nonlinear dynamics.

### DeePC

Data-driven approaches to MPC date back to *dynamic matrix control* [57], [58] and the previously reviewed SPC [49], [50]. We refer to [59], [60], and [61] for recent surveys and contemporary learning-based MPC approaches. Here, our starting point is the formulation (30) proposed for deterministic LTI systems in [23].

In the case of inexact data or non-LTI dynamics, it is tempting to opt for certainty equivalence: take the data at face value and implement the control (30). However, this approach is doomed to fail. This can be intuitively understood from the perspective of dictionary learning alluded to in Figure 6. The optimal control trajectory is synthesized by linearly combining columns of the Hankel matrix $\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}(w_{\text{d}})$, which spans $\mathscr{B}|_{T_{\text{ini}}+T_{\text{f}}}$. However, a superposition of trajectories from $\mathscr{B}|_{T_{\text{ini}}+T_{\text{f}}}$ is, again, a valid trajectory of $\mathscr{B}|_{T_{\text{ini}}+T_{\text{f}}}$ only for deterministic LTI systems. What is even more detrimental in a practical setting is that a Hankel matrix $\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}$ built from noisy or nonlinear data will almost surely have full rank, and, thus, any future trajectory $(u_{\text{f}}, y_{\text{f}})$ is feasible for (30). As a result, the predicted trajectory can be arbitrarily optimistic and nonrealizable when implemented on the real system. Further, the data $(u_{\text{ini}}, y_{\text{ini}})$ collected online [before implementing an instance of the optimal control (30)] is typically noise corrupted as well, which leads to further deterioration of the realized performance and even feasibility issues.

For these reasons, a robust approach is favored over certainty equivalence. *DeePC* [62] is a robustified version of direct data-driven LQ control (30)

$$\begin{aligned}\text{minimize} \quad & \text{over } u_{\text{f}}, y_{\text{f}}, g, \sigma_{u_{\text{ini}}}, \sigma_{y_{\text{ini}}} \quad \|y_{\text{f}} - y_{\text{r}}\|_Q^2 + \|u_{\text{f}} - u_{\text{r}}\|_R^2 \\ & + \lambda_{u_{\text{ini}}} \|\sigma_{u_{\text{ini}}}\|_2^2 + \lambda_{y_{\text{ini}}} \|\sigma_{y_{\text{ini}}}\|_2^2 + \lambda_g \cdot h(g) \\ \text{subject to} \quad & \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \\ Y_{\text{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} + \sigma_{u_{\text{ini}}} \\ y_{\text{ini}} + \sigma_{y_{\text{ini}}} \\ u_{\text{f}} \\ y_{\text{f}} \end{bmatrix} \text{ and } (u_{\text{f}}, y_{\text{f}}) \in \mathscr{U} \times \mathscr{Y}. \end{aligned} \tag{31}$$

Let us walk slowly through the DeePC problem formulation (31).

First, recall that $w_{\text{ini}} = (u_{\text{ini}}, y_{\text{ini}})$ are online measurements used to estimate an initial condition via the constraint equations $U_{\text{p}} g = u_{\text{ini}}$ and $Y_{\text{p}} g = y_{\text{ini}}$ in the certainty-equivalence LQ control (30). However, for inexact data, these constraints may not be feasible. Hence, DeePC (31) opts for least-squares estimation akin to MHE and softens these constraints with the slack variables $\sigma_{u_{\text{ini}}}$ and $\sigma_{y_{\text{ini}}}$, which are penalized in the cost with nonnegative coefficients $\lambda_{u_{\text{ini}}}$ and $\lambda_{y_{\text{ini}}}$. Different variations of the penalty are conceivable, for example, weighting the norms by inverse noise covariances, disregarding $\sigma_{u_{\text{ini}}}$ in absence of

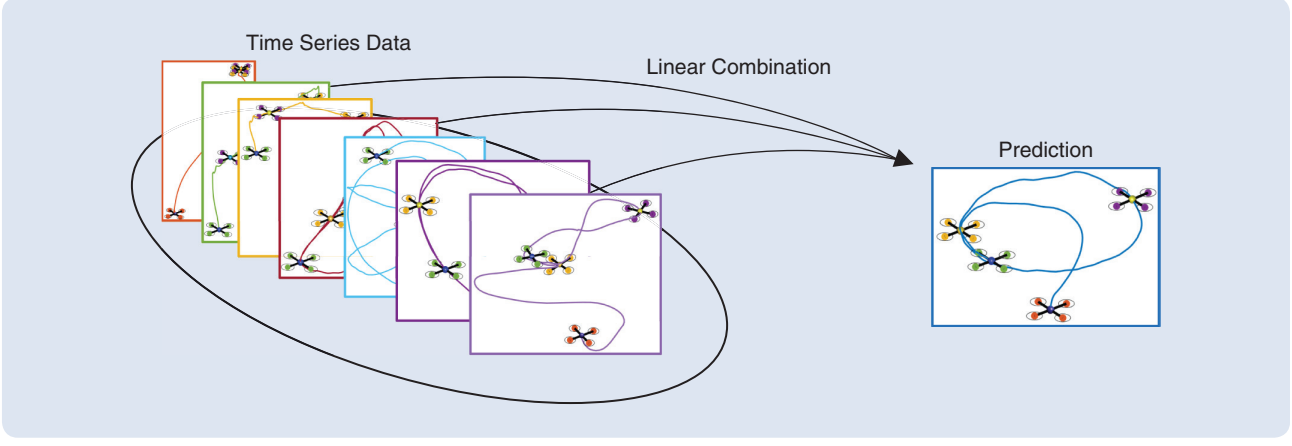



**FIGURE 6** A schematic illustration of the direct data-driven LQ control approach (30) to synthesize the motion of a quadcopter. Trajectories from a library are linearly combined to match the pre-fix trajectory and to optimally synthesize the future control trajectory. We refer to [53] for an implementation of this case study.

input noise, or opting for a linear penalty in the spirit of exact penalization.

A second minor—yet practicably important—modification is to augment the data-driven LQ problem (30) with input and output constraints $u_f \in \mathscr{U}$ and $y_f \in \mathscr{Y}$, respectively. These can account for, for example, input saturation, operational limits, or terminal constraints. The latter are useful for closed-loop stability when implementing (31) as a receding-horizon predictive controller [44], [45]; see Figure 7. We will return to these points later in the "DeePC: Closed-Loop and Implementation Aspects" section.

Third, aside from uncertainty in $(u_{ini}, y_{ini})$ affecting the constraints *additively*, the certainty-equivalence LQ problem (30) is also subject to *multiplicative* uncertainty since the data matrices $U_p$, $Y_p$, $U_f$, and $Y_f$ may also be inexact (noisy or from a nonlinear system). This source of noise can be mitigated offline by preprocessing the trajectory library [for example, by seeking a low-rank approximation of $\mathscr{H}_{T_{ini}+T_f}(w_d)$]. However, in the spirit of direct data-driven control—seeking an online decision based on raw data—DeePC opts for regularizing the problem (30). In particular, a nonnegative term $h(g)$ is added to the cost function with a to-be-tuned regularization coefficient $\lambda_g \geq 0$. Different regularization terms will be derived and motivated later.

Last, it is instructive to recall the involved timescales and datasets when implementing DeePC (31) or any of the approaches (25)–(30). We refer to Figure 7 for a schematic illustration. The data $(u_d, y_d)$ of length $T_d$ are collected offline and directly parameterize the optimal control problem via the matrices $U_p, Y_p, U_f$, and $Y_f$. In a model-based setting, the offline data are used to identify a model. In the online stage, the last $T_{ini}$ measurements $(u_{ini}, y_{ini})$ are used to initialize the optimal control problem and to predict the future system evolution $(u_f, y_f)$ of length $T_f$. Finally, in a receding-horizon implementation, the DeePC problem (31) is repeatedly called upon for estimation and prediction, and the datasets $(u_{ini}, y_{ini})$ and $(u_f, y_f)$ are updated accordingly. Of course, a natural (albeit nontrivial) extension is to update the data matrices online in the spirit of adaptive control.

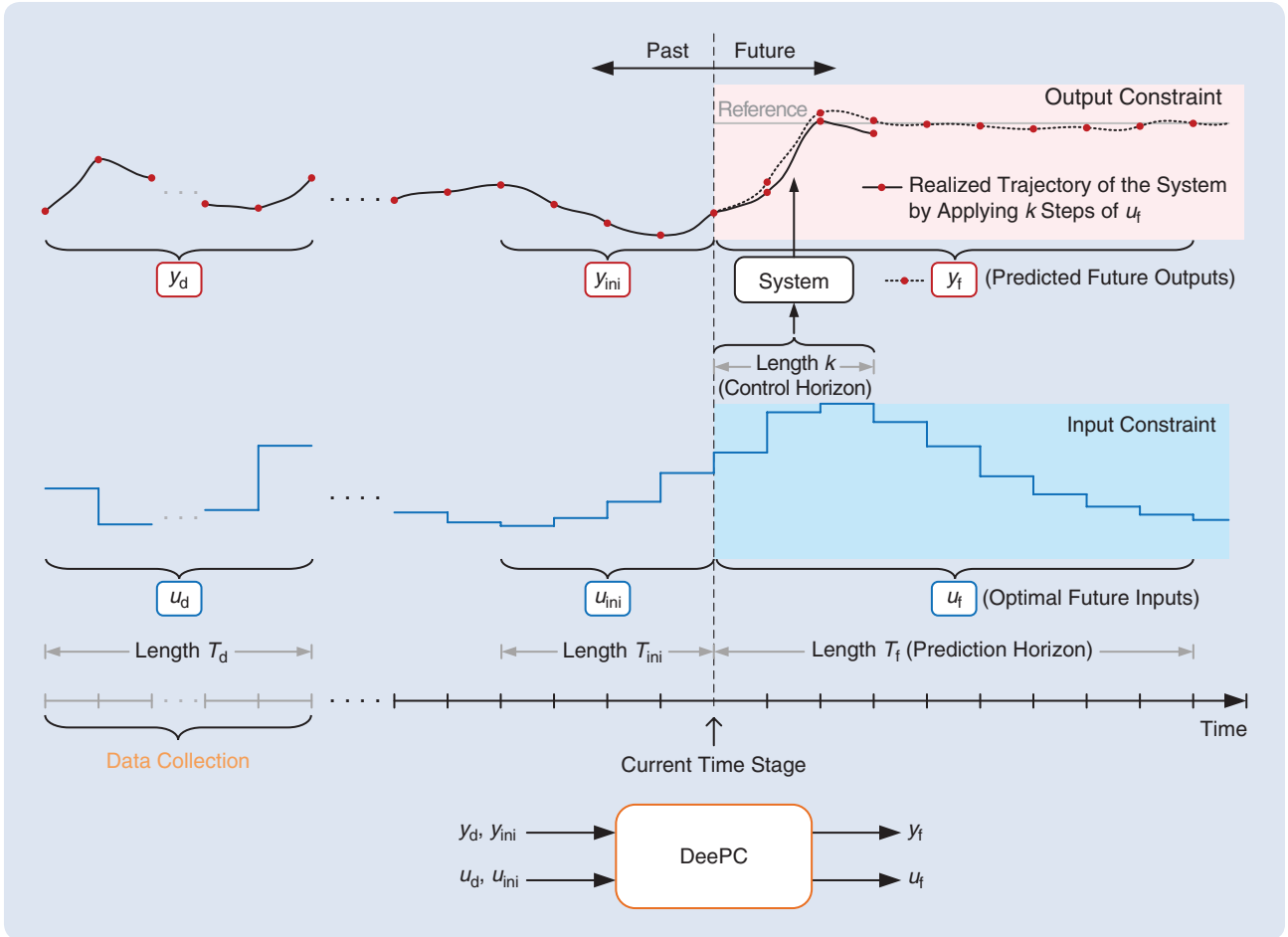



**FIGURE 7** The different timescales and datasets when implementing DeePC (31). In the offline stage, the data $(u_d, y_d)$ of length $T_d$ are collected to build the matrices $U_p, Y_p, U_f$, and $Y_f$. In the online stage, the last $T_{ini}$ measurements $(u_{ini}, y_{ini})$ are used to initialize the optimal control and to predict the future evolution $(u_f, y_f)$ of length $T_f$. In a receding-horizon setting, the DeePC problem (31) is repeatedly called upon after $k$ time steps ($k$ being the control horizon), and the datasets $(u_{ini}, y_{ini})$ and $(u_f, y_f)$ are updated.

### *The Role of Regularization in DeePC*

Observe that the DeePC formulation (31) can be compactified by eliminating the variables $u_f$, $y_f$, $\sigma_{u_{ini}}$, $\sigma_{y_{ini}}$

$$\begin{aligned} \text{minimize} \quad & \text{over } g \quad \| Y_f g - y_r \|_Q^2 + \| U_f g - u_r \|_R^2 \\ & + \lambda_{y_{ini}} \| Y_p g - y_{ini} \|_2^2 + \lambda_{u_{ini}} \| U_p g - u_{ini} \|_2^2 + \lambda_g \cdot h(g) \\ \text{subject to} \quad & (U_f g, Y_f g) \in \mathscr{U} \times \mathscr{Y}. \end{aligned} \tag{32}$$

Define further $w_{r,ini} = (u_{ini}, y_{ini}, u_r, y_r)$ and the block-diagonal matrix $P = \text{blkdiag}(\lambda_{u_{ini}} I, \lambda_{y_{ini}} I, R, Q)$. Then, in absence of constraints, (32) takes the form of a regularized regression problem

$$\text{minimize} \quad \text{over } g \quad \| \mathscr{H}_{T_{ini}+T_f}(w_d) g - w_{r,ini} \|_P^2 + \lambda_g \cdot h(g). \tag{33}$$

The compact formulation (33) provides a regression perspective on DeePC illustrated in Figure 8, and it motivates the application of Bayesian, nonparametric, or robust regression methods. In particular, we can draw from the vast literature on regularized regression to derive, inspire, or interpret different regularizers $h(g)$. "Roles of Regularization" explains three often advocated, dominant reasons for why to regularize:

- aiding the optimization by strong convexification
- conditioning models on prior knowledge
- robustifying the problem formulation.

For the DeePC problem (31), different regularizations have first been proposed as heuristics [62] before being constructively derived. In what follows, we present different variations in a tutorial style and refer to [17] for a more detailed account.

#### Regularization Derived From Preprocessing

For inexact data, the matrix $\mathscr{H}_{T_{ini}+T_f}(w_d)$ generically does not have rank $m(T_{ini}+T_f)+n$ and does not reveal an LTI behavior of desired complexity. As previously discussed, the noisy data matrix can be preprocessed via structured low-rank approximation. Formally, this can be posed as a bilevel optimization problem: solve the optimal control problem subject to preprocessing of the data

$$\begin{aligned} \text{minimize} \quad & \text{over } g \quad \| \mathscr{H}_{T_{ini}+T_f}(\hat{w}_d^\star) g - w_{r,ini} \|_P^2 \\ \text{subject to} \quad & \hat{w}_d^\star \in \text{argmin over } \hat{w}_d \quad \| w_d - \hat{w}_d \| \\ \text{subject to} \quad & \text{rank}\, \mathscr{H}_{T_{ini}+T_f}(\hat{w}_d) \le m \cdot (T_{ini}+T_f) + n. \end{aligned}$$

This nonconvex bilevel problem can be formally reduced and convexified. We refer to [11] for details and provide an informal account. Since the processed data matrix $\mathscr{H}_{T_{ini}+T_f}(\hat{w}_d^\star)$ has low rank, $g$ can be chosen to have at most $m \cdot (T_{ini}+T_f)+n$ entries, and one may (without affecting feasibility or optimality) add a cardinality constraint to the outer problem: $\| g \|_0 \le m \cdot (T_{ini}+T_f)+n$. This constraint can be relaxed to an $\ell_1$-norm and lifted to the objective (with sufficiently large penalty), resulting in

$$\begin{aligned} \text{minimize} \quad & \text{over } g \quad \| \mathscr{H}_{T_{ini}+T_f}(\hat{w}_d^\star) g - w_{r,ini} \|_P^2 + \lambda_g \cdot \| g \|_1 \\ \text{subject to} \quad & \hat{w}_d^\star \in \text{argmin over } \hat{w}_d \quad \| w_d - \hat{w}_d \| \\ & \text{subject to} \quad \text{rank}\, \mathscr{H}_{T_{ini}+T_f}(\hat{w}_d) \le m \cdot (T_{ini}+T_f) + n \end{aligned}$$

for $\lambda_g \ge 0$ sufficiently large. Next, the inner problem is relaxed by dropping the rank constraint so that $\hat{w}_d^\star = w_d$; that is, the data are taken at face value. Hence, we arrive at the regularized problem (33) with $h(g) = \| g \|_1$. In summary, an $\ell_1$-norm regularizer can be interpreted as a surrogate for preprocessing the data via a low-rank approximation to select the model complexity.

Of course, such a problem relaxation allows for a larger solution space, including trajectories that may not be consistent with the plant behavior. In the authors' experience stemming from manifold implementations, such relaxed solutions may still lead to excellent performance and may even out-compete (local) solutions targeting the nonconvex bilevel problem (see [11] and [31]—provided that the coefficient $\lambda_g$ is suitably selected, which will be a recurring point throughout the remainder of this article.

#### Regularization Derived From Least-Squares SysID

Recall the indirect LQ control (29). Note that the explicit multistep predictor (27)–(28) can be written as $y_f = Y_f g$, with $g$ satisfying

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u_f \end{bmatrix} \quad \text{and} \quad 0 = \left\| \left( I - \underbrace{\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}}_{=:\Pi} \right) g \right\|_p. \tag{34}$$

In (34), we identify the first three constraint equations of the data-driven LQ problem (30) and an orthogonality constraint with the projector $\Pi$ ensuring that a minimum-norm solution $\| g \|_p$ (for any $p \in [1, \infty]$) to these equations is sought. Thus, for exact data, without affecting feasibility or optimality, the regularizer

$$h(g) = \| (I - \Pi) g \|_p \tag{35}$$

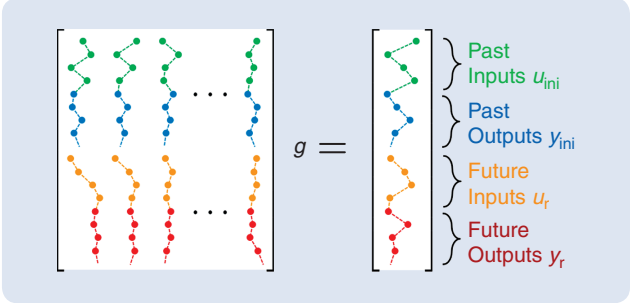



**FIGURE 8** A schematic illustration of the DeePC formulation (33) as a regression problem. The vector $g$ robustly selects elements from the trajectory library span $\mathscr{H}_{T_{ini}+T_f}(w_d)$ to match the past pre-fix trajectory $(u_{ini}, y_{ini})$ as well as the future reference trajectory $(u_r, y_r)$. Robustness is enforced by choosing the regularizer in (33).

## Roles of Regularization

In what follows, we provide an intuition of the different roles that regularization plays and which objectives it can achieve by revisiting classic least-squares regression akin to (20) or (33).

Consider a set of $n$ data pairs $\{x_i, y_i\} \in \mathbb{R}^p \times \mathbb{R}, i \in \{1, \ldots, n\}$ denoting independent and dependent variables, that is, inputs and outputs. We aim to relate these by a linear model $y_i = x_i^\top \theta$ and seek to estimate a suitable parameter vector $\theta \in \mathbb{R}^p$. By stacking all measurements, we obtain $Y = X\theta$, where $X_{ij}$ denotes the $j$th entry of the $i$th measurement, and $Y = [y_1 \ldots y_n]^\top$.

If we assume that the output measurements are subject to additive unit-variance Gaussian noise, then a maximization of the log-likelihood function (that is, the likelihood of observing the data given the parameters) leads to the classic estimation criterion of minimizing the residual sum of squares [S18]

$$\text{minimize}_\theta \, \| Y - X\theta \|_2^2. \tag{S13}$$

A unique least-squares solution $\theta^\star = (X^\top X)^{-1} X^\top Y$ exists, provided that $X$ has a full column rank. If that is not the case, then a standard modification to the optimization objective (S13) is Tikhonov (or ridge) regularization, strongly convexifying the optimization problem (S13) by adding a quadratic penalty [S19]:

$$\text{minimize}_\theta \, \| Y - X\theta \|_2^2 + \lambda \| \theta \|_2^2. \tag{S14}$$

For a positive coefficient $\lambda > 0$, a unique solution to (S14) of the form $\theta^\star = (X^\top X + \lambda I)^{-1} X^\top Y$ always exists. Here, regularization acts as a patch to literally regularize an otherwise singular matrix $X^\top X$ and to stabilize numerical optimization algorithms. However, regularization can also be derived from principled arguments. Two classic derivations are given here. They are both motivated from the fact that, in the case of noisy data, the solution of the least-squares problem (S13) might be overfitting the dataset, and its solution $\theta^\star$ may perform poorly out of sample (that is, when applying the estimated model $y = x^\top \theta^\star$ to unseen data). Hence, either side information (prior knowledge) must be incorporated, or the estimation criterion has to be roubstified.

In a Bayesian setting, the parameter vector $\theta$ is explicitly regarded as a random variable, and an a prior belief might be that all parameters $\theta_i$ are distributed according to a Gaussian with zero mean and variance $1/\lambda^2$. Then, the maximum a posteriori probability estimate can be obtained by solving the regularized problem (S14) [S18]. In other words, conditioning the regression (S13) on a prior belief about the model results in a regularization.

In a robust estimation setting, one may go beyond additive uncertainty and also consider multiplicative uncertainty affecting the measurement equation $Y = X\theta$, that is, $X$ is noise corrupted. Then, a robust version of the least-squares problem (S13) is

$$\min_\theta \max_{\Delta : \|\Delta\|_F \le \rho} \| Y - (X + \Delta)\theta \|_2 \tag{S15}$$

with compact uncertainty set $\{\Delta : \|\Delta\|_F \le \rho\}$. By the triangle inequality, the inner maximization can be upper-bounded as

$$\begin{aligned} &\max_{\Delta : \|\Delta\|_F \le \rho} \| Y - (X + \Delta)\theta \|_2 \\ &\le \max_{\Delta : \|\Delta\|_F \le \rho} \| Y - X\theta \|_2 + \| \Delta\theta \|_2 = \| Y - X\theta \|_2 + \rho \cdot \| \theta \|_2 \end{aligned}$$

where the last equality follows from submultiplicativity. Finally, note that the triangle inequality is tight within the uncertainty set; that is, when $\Delta$ is chosen so that $Y - X\theta$ and $\Delta\theta$ are collinear

$$\Delta = \frac{\rho}{\| \theta \|} w \theta^\top \text{ where } w = \begin{cases} \dfrac{Y - X\theta}{\| Y - X\theta \|} & \text{if } Y \neq X\theta \\ \text{any unit norm vector} & \text{otherwise.} \end{cases}$$

Thus, the robust least-squares problem (S15) is equivalent to a regularized least-squares problem

$$\text{minimize}_\theta \| Y - X\theta \|_2 + \rho \cdot \| \theta \|_2.$$

Similarly, other regularizers (such as the popular $\ell_1$-regularizer $\| \theta \|_1$) can be derived from either Bayesian perspectives [S18] or (distributionally) robust problem formulations [S20], [S21].

In summary, regularizations 1) aid the optimization by rendering the solution unique, 2) condition models on prior knowledge, and 3) robustify problem formulations. These conceptual insights apply beyond the considered regression problem, for example, in SysID [S22], [10] or in data-driven control [25], [39], [56], [63], [64], as elaborated upon in the main text.

### REFERENCES


[S18] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. New York, NY, USA: Springer, 2009.

[S19] D. Bertsekas, *Convex Optimization Algorithms*. Belmont, MA, USA: Athena Scientific, 2015.

[S20] D. Bertsimas and M. S. Copenhaver, "Characterization of the equivalence of robustification and regularization in linear and matrix regression," *Eur. J. Oper. Res.*, vol. 270, no. 3, pp. 931–942, Nov. 2018, doi: 10.1016/j.ejor.2017.03.051.

[S21] S. Shafieezadeh-Abadeh, D. Kuhn, and P. M. Esfahani, "Regularization via mass transportation," *J. Mach. Learn. Res.*, vol. 20, no. 103, pp. 1–68, Jun. 2019.

[S22] G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, and L. Ljung, *Regularized System Identification: Learning Dynamic Models from Data*. Cham, Switzerland: Springer, 2022.

can be added to problem (30): indeed, for perfect data, all feasible solutions $g$ to (30) result in the same control sequence $(u_f, y_f)$, and the regularizer merely selects a particular solution. However, for inexact data, this regularizer promotes selecting the same solution as the bilevel (SysID followed by control) problem (29). Hence, unlike norm-based regularizers $\|g\|_p$, the projection $I-\Pi$ ensures consistency: it does not bias the solution obtained for perfect data. We refer to [11] for further details, to [56] for an alternative derivation, and to [39] and [63] for an application of the regularizer to the infinite-horizon LQ regulator.

These two regularizers can be interpreted in the Bayesian sense (see "Roles of Regularization"); that is, they condition the data-driven optimal control problem on prior knowledge: namely, they are both derived from a hidden indirect problem formulation, where the data are either preprocessed according to a linear model or such a model is identified via least squares. The magnitude of the regularization coefficient $\lambda_g$ ensures to which extent the inner preprocessing/identification problems are (approximately) enforced. However, unlike in the indirect formulations, no projection on the class of LTI models of desired complexity is enforced. As a result, the noise is not entirely removed (no variance reduction), but no erroneous model selection (no bias) is encountered. These bias–variance tradeoff discussions give an intuition when indirect data-driven control approaches are inferior (respectively, superior) to the direct DeePC formulation; see [11] and [65] for further elaborations.

### Regularizations Derived From Robust Optimization

One may also take the robust min-max regression perspective presented in (S15) to robustify the certainty-equivalence LQ problem (30) as

$$\text{minimize over } g \quad \text{maximize over } \hat{w}_\mathrm{d} \in \mathbb{W}(w_\mathrm{d}) \quad \|\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(\hat{w}_\mathrm{d})g - w_\mathrm{r,ini}\|_p^2$$

where $\mathbb{W}(w_\mathrm{d})$ is an uncertainty set typically centered at the collected offline data $w_\mathrm{d}$. Similar to the triangle inequality manipulation following (S15), one can show equivalence of

## Distributional Robustness in Trajectory Space

In the regression formulation of DeePC (33), the columns (or rows) of the Hankel matrix $\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(\hat{w}_\mathrm{d})$ are trajectories, and the objective is to minimize the loss by linearly combining them; see Figure 8. We now take a stochastic perspective and consider each trajectory as an observed sample from some unknown stochastic process. Then, we can equivalently reformulate (33) (without the regularization term) as

$$\text{minimize} \quad \text{over } g \quad \mathbb{E}_{\hat{w}_\mathrm{d}\sim\hat{\mathbb{P}}}[\|\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(\hat{w}_\mathrm{d})g - w_\mathrm{r,ini}\|_p^2] \tag{S16}$$

where $\hat{\mathbb{P}}$ is the associated empirical distribution built using the data samples $w_\mathrm{d}$; that is, the measure of $\hat{w}_\mathrm{d}$ is supported on $w_\mathrm{d}$.

When implementing the solution of the so-called sample average approximation (S16) on the real system, one should expect a poor out-of-sample performance since the true data-generating distribution is due to some (possibly nonlinear, non-Gaussian) stochastic process that is only poorly represented by the sample distribution $\hat{\mathbb{P}}$. To robustify against such processes, [25], [74] propose the *distributionally robust formulation*

$$\inf_{g\in\mathscr{G}} \sup_{\mathbb{Q}\in\mathbb{B}_\varepsilon(\hat{\mathbb{P}})} \mathbb{E}_{\hat{w}_\mathrm{d}\sim\mathbb{Q}}[\|\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(\hat{w}_\mathrm{d})g - w_\mathrm{r,ini}\|_p^2] \tag{S17}$$

where the expectation is taken with respect to an unknown distribution $\mathbb{Q}$ taking value in a so-called *ambiguity set* of all distributions that are $\varepsilon$-close to the sample distribution $\hat{\mathbb{P}}$. Problem (S17) seeks to be robust to all such distributions $\mathbb{Q}$. Hence, the terminology "distributional robustness." More precisely, the ambiguity set is a *Wasserstein ball* of radius $\varepsilon > 0$, centered at $\hat{\mathbb{P}}$, and denoted by $\mathbb{B}_\varepsilon(\hat{\mathbb{P}})$. Let us elaborate on these ingredients from a high-level and accessible perspective.

More formally, let $\mathscr{M}(\Xi)$ be the set of all distributions $\mathbb{Q}$ supported on $\Xi$ (a subset of a vector space), such that $\mathbb{E}_\mathbb{Q}[\|\xi\|_p] < \infty$, where $\|\cdot\|_p$ is the $p$-norm for some $p \in [1,\infty]$. The *Wasserstein distance* $\mathrm{d_W}\colon \mathscr{M}(\Xi)\times\mathscr{M}(\Xi) \to \mathbb{R}_{\geq 0}$ between two distributions $\mathbb{Q}_1$ and $\mathbb{Q}_2$, each supported on $\Xi$, is

$$\mathrm{d_W}(\mathbb{Q}_1,\mathbb{Q}_2) = \inf_\Pi \left\{ \int_\Xi \int_\Xi \|\xi_1 - \xi_2\|_p \mathrm{d}\Pi(\xi_1\xi_2) \right\} \tag{S18}$$

where $\Pi$ is the set of distributions over $\mathscr{M}(\Xi\times\Xi)$ with marginal distributions $\mathbb{Q}_1 \in \mathscr{M}(\Xi)$ and $\mathbb{Q}_2 \in \mathscr{M}(\Xi)$, respectively.

The semantics of (S18) are as follows: we seek the minimum expected cost to transport the probability distribution $\mathbb{Q}_1$ onto the probability distribution $\mathbb{Q}_2$ when transporting a unit of mass from $\xi_1$ to $\xi_2$ costs $\|\xi_1 - \xi_2\|_p$. We refer to Figure S5 for a graphical illustration. Hence, the Wasserstein distance is colloquially also referred as the Earth mover's distance. While there are many potential transport plans $\Pi$ that map $\mathbb{Q}_1$ onto $\mathbb{Q}_2$, the Wasserstein distance seeks the minimum cost solution.

Finally, given the distance (S18), we denote the *Wasserstein ball* of radius $\varepsilon \geq 0$ centered around a distribution $\mathbb{Q}$ by

$$B_\varepsilon(\mathbb{Q}) = \{\mathbb{Q}' \in \mathscr{M}(\Xi) \mid d_\mathrm{W}(\mathbb{Q},\mathbb{Q}') \leq \varepsilon\}.$$

The Wasserstein distance has intimate connection to optimal transport [S23], [S24], and, unlike other distances (or

the inner maximization and a regularization. Different uncertainty sets give rise to different regularizers. One may consider structured and unstructured uncertainty sets ranging from mere norm balls, over interval-valued and columnwise uncertainties, to uncertainties with the Hankel structure. For each of these, [64] and [66] propose tractable reformulations, many of which result in regularization terms $h(g)$ taking the form of weighted $p$-norms: for example, a uniform columnwise uncertainty (for example, relevant for a trajectory matrix data structure) results in $h(g) = \| g \|_1$.

A similar (albeit stochastic) perspective leading to regularization is due to distributional robustness; see "Distributional Robustness in Trajectory Space" for details. Last, we also remark that regularization is a key aspect in the closed-loop stability and robustness analysis of DeePC [67].

Aside from regularizations, alternative robustifications of the certainty-equivalence LQ problem (30) have been proposed. We mention (among others) [68] and [69], presenting a data-driven formulation of system-level and input/output parameterizations [70], [71] by means of the representation (13). Further, [72] directly robustifies the problem formulation (30) to adversarial and measurable disturbances. Finally, [73] proposes a maximum-likelihood approach to DeePC. We refer to [17] for a detailed overview discussing and comparing different approaches.

### *DeePC: Closed-Loop and Implementation Aspects*

So far, we have discussed DeePC (31) as an open-loop optimal control problem. We now delve into different aspects when implementing it in closed-loop receding-horizon control. In particular, we discuss aspects related to stability, nonlinearity, feasibility, the trajectory library, and computation. We conclude by providing tuning recommendations that proved useful in different domains; see "Selected Implementations of DeePC."

#### Stability

First and foremost, we discuss closed-loop stability and robustness in the presence of noisy data. The sequence of articles [27], [76], [77], [78], and [79] by Berberich and coworkers has shown that many stability and robustness certificates

divergences), it can treat empirical and continuous distributions within a unified framework, which is utterly relevant in a data-driven context. Finally, it is an analytically and computationally tractable metric in a distributional robust optimization problem as (S18) [S25], [S26].

To return to (S17), it ensures robustness to all time series that are $\varepsilon$-close to the data samples $\hat{w}_\mathrm{d}$. Hence, while the problem formulation is intuitive—seeking probabilistic robustness in the space of trajectories—it is an arguably hard semi-infinite problem. Recently, [S21] has shown that, under suitable assumptions, distributionally robust regression problems can be reformulated as regularizations of the sample average approximation. When adopting these methods to (S17), [25] has shown that, when the objective admits a Lipschitz constant $L$, (S17) is equivalent to a version of regularized DeePC (32)

$$\text{minimize over } g \ \| \mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})g - w_\mathrm{r,ini} \|_P + L\varepsilon \cdot \| g \|_q$$

where $1/p + 1/q = 1$; that is, $\| \cdot \|_q$ is the dual norm of the one used to construct the Wasserstein ball. For example, safeguarding against uncertainty in $\ell_\infty$-norm in the space of trajectories is equivalent to $\ell_1$-norm regularization. The radius $\varepsilon$ of the ambiguity can be estimated (albeit conservatively) from the number of data samples, and it can be reduced by averaging datasets.

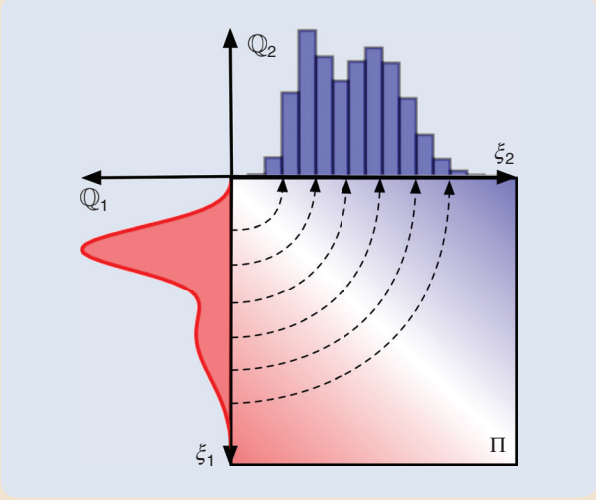



**FIGURE S5** A schematic illustration of the Wasserstein distance between two distributions. Here, $\mathbb{Q}_1$ is an absolutely continuous distribution, $\mathbb{Q}_2$ is an empirical distribution, and $\Pi$ is a joint distribution (or a transport plan) with marginals $\mathbb{Q}_1$ and $\mathbb{Q}_2$.

#### REFERENCES


[S23] C. Villani, *Optimal Transport: Old and New*, vol. 338. Berlin, Germany: Springer-Verlag, 2009.

[S24] G. Peyré and M. Cuturi, "Computational optimal transport: With applications to data science," *Found. Trends Mach. Learn.*, vol. 11, nos. 5–6, pp. 355–607, Feb. 2019, doi: 10.1561/2200000073.

[S25] D. Kuhn, P. Esfahani, V. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," in *Proc. Oper. Res. Manage. Sci. Age Analytics*, Catonsville, MD, USA: INFORMS, 2019, pp. 130–166.

[S26] J. Blanchet, J. Karthyek Murthy, and V. Anh Nguyen, Statistical analysis of Wasserstein distributionally robust estimators," in *Proc. Tut. Oper. Res., Emerg. Optim. Methods Model. Techn. Appl.*, Catonsville, MD, USA: INFORMS, 2021, pp. 227–254.

for MPC carry over to the data-driven domain. However, particular care must be taken when addressing, for example, noise in the data matrix $\mathscr{H}_{T_{\text{ini}}+T_f}(w_d)$ and terminal ingredients that are useful for stability and normally computed from model information. We refer to [80] for a recent tutorial and give a brief self-contained exposition here.

The initial work [67] certifies the closed-loop practical exponential stability of DeePC (31) with input constraints and when applied to noisy LTI systems. The analysis assumes terminal equality constraints, and the proof relies on a nonminimal state-space realization consisting of inputs, outputs, and their shifts as well as an input/output-to-state stability Lyapunov function since the state is not directly penalized in the cost. Further, when certifying robust stability in the noisy case, [67] considers regularizations as in (31), where $h(g) = \|g\|_2^2$ is a ridge regularizer—though the analysis also carries through when using the projection-based regularizer in (35). (This was confirmed in personal communication with the authors of [67].) Since terminal equality constraints are often not desired in practice (for example, due to aggressive behavior or fragile feasibility), the analysis has been extended to terminal equality constraints with an artificial setpoint [27], terminal costs and constraints computable directly from data [77], and implementations without terminal ingredients [78]. The last case without terminal ingredients is particularly useful in practice, as terminal ingredients are hard to design, sometimes result in numerical challenges, and often result in unsatisfactory performance. Further notable extensions address robust output constraint satisfaction [79], time-varying references [76], linear tracking control for affine and nonlinear systems [27], and offset-free tracking via incremental inputs [81].

### *Nonlinearity*

To remain on the theme of nonlinearity, the DeePC method is founded on LTI systems: an optimal trajectory is synthesized by linearly (albeit robustly) combining trajectories from a library according to the superposition principle. However, many case studies (see "Selected Implementations of DeePC") show that DeePC performs remarkably well in a closed loop with weakly nonlinear

## Selected Implementations of DeePC

The simplicity of DeePC and its remarkable empirical performance led to many recent experimental and computational case studies across different domains. For example, demonstrations within robotics include a quadcopter [53]; an autonomous walking excavator [S27]; leading cruise control [S28], [S29]; and a quadruped [S30] next to other small-scale laboratory implementations; see [17] for a survey. Within the energy domain, DeePC has been implemented on grid-connected power converters [66]; synchronous motor drives [S31], [S32]; building automation [75], [S33], [S34]; and battery charging [S35]. Further, [72] provides a decentralized DeePC implementation for power system oscillation damping, and [S36] uses DeePC to control a combined-cycle power plant. At the time of writing, the latter two implementations are being further developed by industrial R&D groups. Concerning the process control domain, an application to a four-tank system is found in [S37]. Finally, yet another intriguing implementation was in greenhouse automation [S38], [S39]. Further applications of DeePC are regularly emerging. The "Application to Control of Power Electronics-Dominated Power Systems" section presents further case studies.

#### REFERENCES


[S27] F. Wegner, J. Coulson, M. Hudoba de Badyn, J. Lygeros, and S. Trimpe, "Data-enabled predictive control of a 12t excavator," Master thesis, ETH Zurich, Zurich, Switzerland, 2021. [Online]. Available: https://www.research-collection.ethz.ch/

[S28] J. Wang et al., "Experimental validation of deep-LCC for dissipating stop-and-go waves in mixed traffic," 2022, *arXiv:2204.03747*.

[S29] J. Wang, Y. Lian, Y. Jiang, Q. Xu, K. Li, and C.N. Jones, "Distributed deep-LCC for cooperatively smoothing large-scale mixed traffic flow via connected and automated vehicles," 2022, *arXiv:2210.13171*.

[S30] R. T. Fawcett, K. Afsari, A. D. Ames, and K. A. Hamed, "Toward a data-driven template model for quadrupedal locomotion," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7636–7643, Jul. 2022, doi: 10.1109/LRA.2022.3184007.

[S31] P. Carlet, A. Favato, S. Bolognani, and F. Dörfler, "Data-driven predictive current control for synchronous motor drives," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2020, pp. 5148–5154, doi: 10.1109/ECCE44975.2020.9235958.

[S32] P. Carlet, F. A. S. Bolognani, and F. Dörfler, "Data-driven continuous-set predictive current control for synchronous motor drives," *IEEE Trans. Power Electron.*, vol. 37, no. 6, pp. 6637–6646, Jun. 2022, doi: 10.1109/TPEL.2022.3142244.

[S33] J. Schwarz, F. Micheli, M. Hudoba de Badyn, and R. Smith, "Data-driven control of buildings and energy hubs," Semester thesis, ETH Zurich, Zurich, Switzerland, 2019. [Online]. Available: https://www.research-collection.ethz.ch/

[S34] L. Di Natale, Y. Lian, E.T. Maddalena, J. Shi, and C.N. Jones, "Lessons learned from data-driven building control experiments: Contrasting gaussian process-based MPC, bilevel DeePC, and deep reinforcement learning," in *Proc. 61st IEEE Conf. Decis. Contr.*, 2022, pp. 1111–1117, doi: 10.1109/CDC51059.2022.9992445.

[S35] K. Chen et al., "Data-enabled predictive control for fast charging of lithium-ion batteries with constraint handling," 2022, *arXiv:2209.12862*.

[S36] P. Mahdavipour, C. Wieland, and H. Spliethoff, "Optimal control of combined-cycle power plants: A data-enabled predictive control perspective," in *Proc. IFAC Conf. Netw. Syst.*, 2022, pp. 103–106.

[S37] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control: Closed-loop guarantees and experimental results," *Automatisierungstechnik*, vol. 69, no. 7, pp. 608–618, Jun. 2021.

[S38] L. Kerkhoff, "Optimal control of autonomous greenhouses: A data-driven approach," Master thesis, TU Delft, Delft, The Netherlands, 2020. [Online]. Available: https://repository.tudelft.nl/islandora/search/

[S39] S. Hemming, F. de Zwart, A. Elings, A. Petropoulou, and I. Righini, "Cherry tomato production in intelligent greenhouses – Sensors and AI for control of climate, irrigation, crop yield, and quality," *Sensors*, vol. 20, no. 22, Nov. 2020, Art. no. 6430.

systems and often outperforms approaches based on linear SysID. To provide a first explanation for this remarkable empirical performance, recall that regularizations enforce robustness in the space of trajectories. Hence, regularized DeePC is certainly robust to nonlinearity. A second argument brought forward in [11] is that the DeePC formulation (31) does not project the data on the class of LTI systems of certain order. Hence, unlike indirect data-driven methods, it cannot encounter a bias error. Third, one may also take the pragmatic perspective that (sufficiently large) LTI systems approximate nonlinear systems well on a finite time horizon, which is theoretically supported by lifting arguments, such as Koopman or Carleman (bi)linearization. Alternatively, one may seek generalizations of the fundamental lemma for classes of nonlinear systems; see [17] for a review. In either case, so far, only [27] has provided theoretic certificates when applying DeePC (with a sliding data window) in a closed loop with a slowly varying nonlinear system.

### *Constraints and Feasibility*

We have already touched upon terminal constraints ensuring closed-loop stability. We now revisit the other constraints in (31). The estimation constraints, $U_\mathrm{p} g = u_\mathrm{ini}$ and $Y_\mathrm{p} g = y_\mathrm{ini}$, have been softened within DeePC (31) by means of slack variables. Instead of such a finite-horizon estimation, one may also adopt a Kalman filtering approach [82] to recursively construct an estimate of a hidden state. Regarding output constraints $Y_\mathrm{f} g \in \mathscr{Y}$, observe that these are based on observed (and, thus, likely inexact) data. Hence, to literally be on the safe side, these constraints have to be suitably robustified and tightened. For instance, [64] considers robustified constraints and provides bounds on the realized system performance. The same methods can also be applied to distributionally robustify stochastic output constraints [25]. Finally, [79] provides closed-loop stability and feasibility certificates for tightened constraints.

#### Trajectory Library

As discussed previously, the data matrix $\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})$ can take a Hankel or trajectory structure. Whereas trajectory matrices can be formed by repeated short experiments (each column equals one experiment), Hankel matrices can be formed by a single long experiment and require considerably fewer data than trajectory matrices. Other data structures fall in between, such as Page or mosaic Hankel matrices. Depending on the application of interest, either data collection approach can be favorable. On the theory side, the main advantages of the different matrix data structures are that the data are conditioned on time invariance (for Hankel matrices) or that columns are independent (for trajectory or Page matrices). The former narrows the hypothesis space for optimal trajectories, which is advantageous in the case of inexact data. The latter allows for less structured preprocessing (for example, low-rank approximation by singular-value thresholding) or robustification (for example, columnwise uncertainties can be considered, resulting in weighted $\ell_1$-regularizations). Finally, empirical results often show a superior realized performance when choosing a less structured trajectory or Page matrices [25], [64], [72].

#### Computational Implementation

To efficiently implement DeePC, the Operator Splitting solver for Quadratic Program (OSQP) solver [83] proved to be a formidable choice. A comparison of the compactified DeePC formulation (31) and the model-based predictor (26) shows that DeePC contains more decision variables [even when eliminating some, as in (32)], its constraints are not sparse, and they have no Toeplitz structure. The computational complexity of DeePC can be reduced by considering low-rank decompositions [56], explicit solutions as in (S12), or multiple shooting formulations [84]. Nevertheless, DeePC is not as computationally efficient as model-based formulations. This insight can also be found for other direct data-driven methods and reveals a distinct advantage of models: they are compressed, denoised, and tidied-up representations.

#### Tuning of Hyperparameters

We highlight the simplicity of the DeePC implementation: similar to MPC and MHE, a cost function, estimation horizon $T_\mathrm{ini}$, prediction horizon $T_\mathrm{f}$, and terminal ingredients need to be chosen in DeePC; see [44] and [45] for standard tuning recommendations. However, in comparison to MPC and MHE, no model is needed. The additional complexity comes from collecting the data and tuning the regularizers. We offer some general recommendations that have proven useful in many case studies, and we refer to the next section for explicit examples.

In terms of data collection, the length $T_\mathrm{d}$ of the data time series $w_\mathrm{d}$ has to be sufficiently long to ensure persistency of excitation; see "The Fundamental Lemma." The lower bound for $T_\mathrm{d}$ depends on the system order, which is generally unknown. On the other hand, the more columns the data matrix has, the better since noise averages out [56]. In practice, a sufficiently large $T_\mathrm{d}$ typically is chosen, and the authors often had good experiences with choosing $T_\mathrm{d}$ so that the data matrix $\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})$ is square.

For the estimation penalty, a quadratic cost, $\lambda_{u_\mathrm{ini}} \|\sigma_{u_\mathrm{ini}}\|_2^2 + \lambda_{y_\mathrm{ini}} \|\sigma_{y_\mathrm{ini}}\|_2^2$, with sufficiently large coefficients $(\lambda_{u_\mathrm{ini}}, \lambda_{y_\mathrm{ini}})$ has proven useful in terms of performance and computational efficiency. Recently, bilevel formulations have emerged, solving an inner optimal estimation problem [56], [75].

As for the choice of regularizers and associated coefficients $\lambda_g h(g)$, the projection-based regularizer (35) with sufficiently large coefficient $\lambda_g$ has proven useful, especially for LTI systems where it does not induce any bias. Often, a squared regularizer is preferred by quadratic

programming solvers. A useful data-based selection criterion for tuning the projection-based regularizers is the *Hanke–Raus heuristic* [81], [85]. Norm-based regularizers $h(g) = \| g \|_p$ may perform equally well but are harder to tune: the realized performance improves when increasing $\lambda_g$ beyond a certain threshold, remains constant for a large interval of $\lambda_g$, and then increases again beyond a second threshold. The (fortunately often shallow) minimum has to be found by cross validation or deduced from prior knowledge (for example, from the size of an uncertainty set). As a rule of thumb, an optimal $\lambda_g$ can be practically found by logarithmically increasing $\lambda_g$.

Figure 9 gives a qualitative summary of this discussion, which is often (though not always) encountered in case studies. For example, the projection-based regularizer $\|(I - \Pi)g\|$ sometimes strictly outperforms norm-based regularizers. Further, the realized performance with the projection-based regularizer is not always monotonically decreasing (as indicated in Figure 9) but may display a strict minimum, implying that the SPC solution (obtained for $\lambda_g$ sufficiently large) is suboptimal. The case study in the next section revisits this discussion.

### *Summary*

This section has presented different indirect (that is, model-based) and direct data-driven control methods. The DeePC formulation (31) has been presented as a robust direct data-driven optimal control formulation. The main robustifications are due to softening the estimation constraints and regularizations, accounting for an implicit SysID or promoting robustness in the space of trajectories.

## APPLICATION TO CONTROL OF POWER ELECTRONICS-DOMINATED POWER SYSTEMS

To tackle climate change, countries from all over the world have made ambitious plans to reduce carbon emissions, with the ultimate goal of achieving *net zero*. As a dominant carbon emitter, the power and energy industry plays a vital role toward this goal, where integrating renewable energy sources is one of the primary approaches for decarbonization.

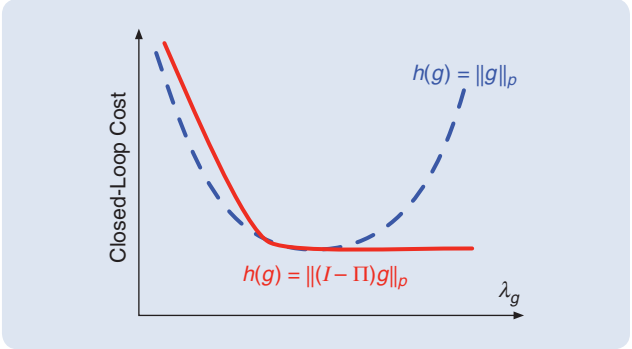



**FIGURE 9** A qualitative illustration of the realized closed-loop performance as a function of the coefficient $\lambda_g$ for a projection-based (red solid) and a norm-based (blue dashed) regularizer. This qualitative behavior has been observed by the authors across many case studies involving LTI systems and weakly nonlinear systems.

Renewable energy sources are distinct from conventional fossil-fuel-based power plants—they have low inertia, suffer from the intermittency of renewables, and are distributed and nondispatchable [86], [87]. Moreover, fossil-fuel-based power plants are connected to the power grid via synchronous generators (SGs), whereas renewable energy sources are connected to the power grid via *power electronics converters* that interconnect dc circuits with ac circuits [88]. For instance, a solar panel produces dc voltage and current, and a power converter is needed to connect the solar panel to the ac power grid; as another example, a direct-drive wind turbine produces ac voltage and current with a variable frequency due to the fluctuation of wind speed, and two back-to-back power converters are needed to connect it to the ac power grid (with fixed frequency), as shown in Figure 10. Moreover, power converters are also widely used as grid interfaces of energy-storage systems, high-voltage dc transmission systems, electric vehicles, and so on. Hence, with more fossil-fuel-based power plants being replaced by renewable energy sources, many countries worldwide are currently witnessing the establishment of *power electronics-dominated power systems*, as shown in Figure 10.

Unlike SGs, power converters have fast actuation and high flexibility in controlling the voltage angle and magnitude. For instance, according to Newton's laws of motion, one can only indirectly change the rotor angle of an SG by changing the torques to accelerate or decelerate the rotor; however, in a power converter, one can directly change the voltage angle by giving a command in the control algorithm. Therefore, it is possible to ensure a resilient power electronics-dominated power system given appropriate control design. However, there are a lot of challenges in designing reliable controllers for grid-connected power converters, as discussed in "Challenges in Control of Power Electronics-Dominated Power Systems."

The dynamics of a grid-connected power converter depends on both the inherent control scheme and characteristics of the power grid. However, on the one hand, the real power grid is complex, variable, and generally unknown from the perspective of a power converter [54]. On the other hand, power converter suppliers usually do not share their proprietary models with the manufacturers of wind generators. Moreover, on a wind farm, there are complex aerodynamic interactions among the wind turbines that are, in general, difficult to model and treated as disturbances [89]. Hence, in practice, it would be cumbersome to do model-based design and manual tuning by commissioning engineers, as they may have neither a model of the grid nor a model of the converter. Nonetheless, the dynamics of a grid-connected wind generator can be captured by the *input and output data* in the controller of the power converter, which interacts with the power grid in a closed-loop manner. These input and output data can be used in data-driven control

design, and one open question is whether direct data-driven methods have advantages over indirect data-driven methods.

In what follows, we apply the DeePC method from the previous section for direct data-driven optimal control in wind generators, which may shed some light on this question. We consider both device-level control (including two scenarios: stabilization of wind generators and grid synchronization and dc voltage regulation) and system-level control (frequency control provided by wind farms) to demonstrate the possibilities and potentials of using data-driven control to ensure a resilient power electronics-dominated power system.

### *Case Study 1: Stabilizing Wind Generators*

The conventional control design of wind generators is model based, which implicitly assumes that the power grid is an ideal one (with fixed voltage magnitude and frequency). However, as discussed in "Challenges in Control of Power Electronics-Dominated Power Systems," the resulting controller may lead to inferior performance or even instabilities when the wind generators are connected to nonideal grids. Although conventional robust and adaptive model-based methods can potentially be used to handle these problems, they may result in complex controllers (with often disappointing performance). Moreover, it is, in general, difficult to find a single fixed controller that is robust to all possible power grid situations.

In what follows, we use DeePC to eliminate oscillations in wind generators, which are caused by the closed-loop interaction between the grid-side power converter and the unknown power grid [90], [91]. Here, we abstract the power grid as a voltage source behind a transmission line, whose impedance is unknown from the perspective of the wind generator. This assumption is mainly for simplicity of presentation and will be lifted in a subsequent case study. Although the DeePC method is general, to illustrate the point, we consider the direct-drive (type 4) wind generator in Figure 11. We use an industrial wind generator model implemented in MATLAB/Simulink (2020b): "Wind Farm—Synchronous Generator and Full-Scale Converter (Type 4) Average Model." This simulation model is of high fidelity and contains details of a wind generator, including the turbine dynamics, flux dynamics, filters, speed control, pitch control, converter control, and maximum power point tracking (MPPT). It has 37 state variables and is highly nonlinear due to the nonlinear power flow equation, MPPT curve, and (adaptive) phase-locked loop (PLL). In industrial practice, this model and its parameters might be proprietary and unknown to commissioning control engineers and system operators of the power grid. It might also be difficult to use conventional SysID methods to obtain a detailed and accurate model due to the high complexity.

To study the oscillation problem, we change the length of the transmission line in the original model to create a weak power grid (that is, a power grid that has long transmission lines and little rotational generation), with the short circuit ratio being 2.23. Note that, in a grid-connected

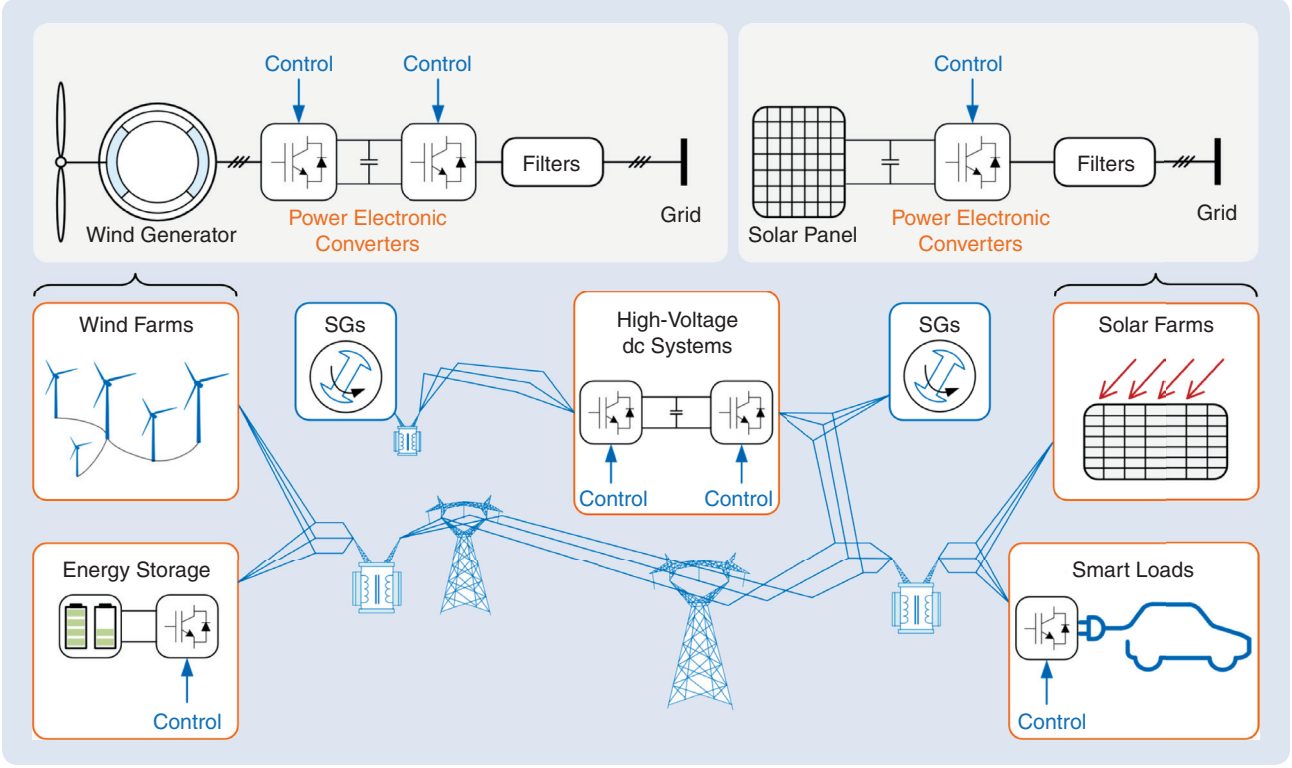



**FIGURE 10** A power electronics-dominated power system.

single-device system, the short circuit ratio is the inverse of the grid impedance [in per-unit (p.u.) values], which characterizes the strength of the power grid [92]. The studied wind generator relies on a synchronous reference frame PLL [93], [94] to detect the voltage phase of the power grid and realize synchronization. We disable the automatic gain control of the PLL in the original model and fix the PLL bandwidth to be 60 rad/s to avoid strongly time-varying dynamics. When connected to such a weak power grid, the wind generator has a low stability margin and can easily become unstable when the power grid changes. To deal with this issue, we employ the DeePC method to perform direct data-driven control and stabilize the system.

As shown in Figure 11, we choose the active power ($P_E$) and reactive power ($Q_E$) to be the two outputs of the unknown system, and DeePC provides two control inputs

## Challenges in Control of Power Electronics-Dominated Power Systems

The large-scale integration of *power electronics converters* into power systems is an inevitable trend, as they act as interfaces between the power grid and renewable sources, electric vehicles, and high-voltage dc transmission systems. In these devices, by performing pulsewidth modulation, one can directly control the sinusoidal voltage angle and magnitude of the converter at the ac side, which provides high flexibility in designing the converters' characteristics and enabling fast control actions [S40].

However, the control design of power converters is a challenging problem, especially in the power system context. On the one hand, due to the fast control, the converter can respond to the disturbances from the power grid in different timescales. However, to maintain the desired performance, the controller should be robust against a variety of uncertainties. On the other hand, in a very fast timescale (milliseconds), the controller needs to be designed and implemented in a fully decentralized manner considering constraints (for example, current and voltage limitations), as fast communication is rarely available in bulk power systems. Moreover, the decentralized design should ensure that the interconnected system has satisfactory performance.

Conventionally, these requirements are handled by employing multiple nested control loops based on proportional-integral-derivative (PID) controllers to achieve power/voltage regulations and grid synchronization in power converters, and the constraints are implemented using saturations and antiwindup strategies [S41], [S42]. The corresponding control structure (that is, how different loops are connected) is designed based on engineering experience, and the PID control gains are tuned manually. Also, lots of effort has been put into the modeling of grid-connected power converters, which provides insights into the system dynamics and criteria for the PID gain tuning. However, detailed and accurate models of such systems are rarely available for control design due to three main reasons:

- Often, power converter manufacturers do not share detailed proprietary models with the control designer of wind turbine manufacturers, and vice versa.
- The power grid, which interacts with the power converter in a closed-loop manner, is generally unknown from the converter's point of view because the grid is ever changing, high dimensional, and subject to lots of uncertainties.
- Finally, exogenous disturbances (for example, the intermittency of renewables) are hard to model, especially on fast timescales.

For simplicity, a presumed model can be used for the control design, which usually assumes that the converter is connected to an ideal source and an ideal grid: a voltage source that has fixed frequency and voltage magnitude. However, this assumption is often not (even close to) true in practice, and, thus, the model-based control design may result in inferior performance or even instabilities.

For instance, many oscillation events (small-signal instabilities) have been observed in real-world wind farms all over the planet, for example, those in the United States and China, especially when the wind farms are integrated in weak power grids [S43]. These oscillation events (which were caused by the mismatch between a presumed model in the control design and the real-world model) resulted in great loss of wind power, as the wind farms had to be disconnected from the grid to protect the wind generators. This challenge can be difficult to address using model-based control design methods. Fortunately, data are readily available during the operation of a power system, which can be used to capture the system dynamics and perform optimal control [72], [S44]. In the "Case Study 1: Stabilizing Wind Generators" section, we demonstrate how direct data-driven control excels in stabilizing wind generators by performing optimal control.

### REFERENCES


[S40] F. Blaabjerg, R. Teodorescu, M. Liserre, and A.V. Timbus, "Overview of control and grid synchronization for distributed power generation systems," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1398–1409, Oct. 2016, doi: 10.1109/TIE.2006.881997.

[S41] L. Huang, H. Xin, and F. Dörfler, "H∞-control of grid-connected converters: Design, objectives and decentralized stability certificates," *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 3805–3816, Sep. 2020, doi: 10.1109/TSG.2020.2984946.

[S42] L. Huang et al., "Transient stability analysis and control design of droop-controlled voltage source converters considering current limitation," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 578–591, Jan. 2019, doi: 10.1109/TSG.2017.2749259.

[S43] Y. Cheng et al., "Wind energy systems sub-synchronous oscillations: Events and modeling," IEEE Power Energy Soc., Piscataway, NJ, USA, Tech. Rep., PES-TR80, 2020.

[S44] A. Chakrabortty, "Wide-area damping control of power systems using dynamic clustering and TCSC-based redesigns," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1503–1514, Sep. 2012, doi: 10.1109/TSG.2012.2197029.

added to the current references. The input/output signals are all in p.u. values. We consider three different regularizers, including a quadratic regularizer $h(g) = \|g\|_2^2$, a one-norm regularizer $h(g) = \|g\|_1$, and a projection-based regularizer $h(g) = \|(I - \Pi)g\|_2^2$. The parameters of DeePC are as follows: $T_d = 600$, $T_{ini} = 6$, $T_f = 12$, $R = I$, $Q = 100I$, and $\lambda_{y_{ini}} = 1{,}000$. We assume no input noise in our case studies and, thus, disregard the slack variable $\sigma_{u_{ini}}$. We choose $\lambda_g = 1$ for the quadratic regularizer and the projection-based regularizer and $\lambda_g = 0.01$ for the one-norm regularizer. Note that the hyperparameters in our case studies are selected manually, and one may achieve better performance by finding a better set of parameters; see Figures 12 and 13. The control horizon is $k = 1$; that is, we apply the first $k$ elements of $u_f$ to the system before solving DeePC again (see Figure 7). The reference trajectory is $r = \mathbf{1} \otimes \left[\begin{smallmatrix} P_s \\ Q_s \end{smallmatrix}\right]$, where $\mathbf{1}$ is the vector of ones; $P_s$ and $Q_s$ are, respectively, the steady-state values of active and reactive power; and $\otimes$ denotes the Kronecker product. Hence, DeePC aims at minimizing the deviations of active and reactive power from their steady-state values, thereby eliminating power oscillations.

The sampling time is 5 ms. In the constraints, we consider the lower and upper bounds of the two inputs to be –0.05 and 0.05, respectively; the lower and upper bounds of the active power (output 1) are 0 and 1.2, respectively; and the lower and upper bounds of the reactive power (output 2) are –1 and 1, respectively. In the following case studies, we observe that these input/output constraints are inactive due to the favorable tracking performance of DeePC. Note that, when the input/output constraints are inactive, one can also derive closed-form solutions of DeePC when the quadratic regularizer or the projection-based regularizer is used [66].

Figure 14 shows the time-domain responses of the wind generator. At *time* = 1 s, we change the short circuit ratio from 2.23 to 2 to emulate an event in the power grid, for example, tripping of transmission lines. It can be seen that the wind generator starts to oscillate after the disturbance, which is caused by the interactions among PLL, current/power control loops, and the weak power grid [90]. After the oscillation is observed, we inject band-limited white noise signals into the system through the two input channels to excite the system and collect data. With the sampling time being 5 ms, we only require 3 s (from *time* = 3 s to *time* = 6 s as displayed in Figure 14) to collect the input/output trajectory of length $T_d = 600$ to construct the Hankel matrices.

One may use other types of persistently exciting input signals to possibly achieve better performance and safe perturbations, for example, a pseudorandom binary sequence. Moreover, one may also avoid actively injecting excitation signals into the system by collecting long enough oscillation

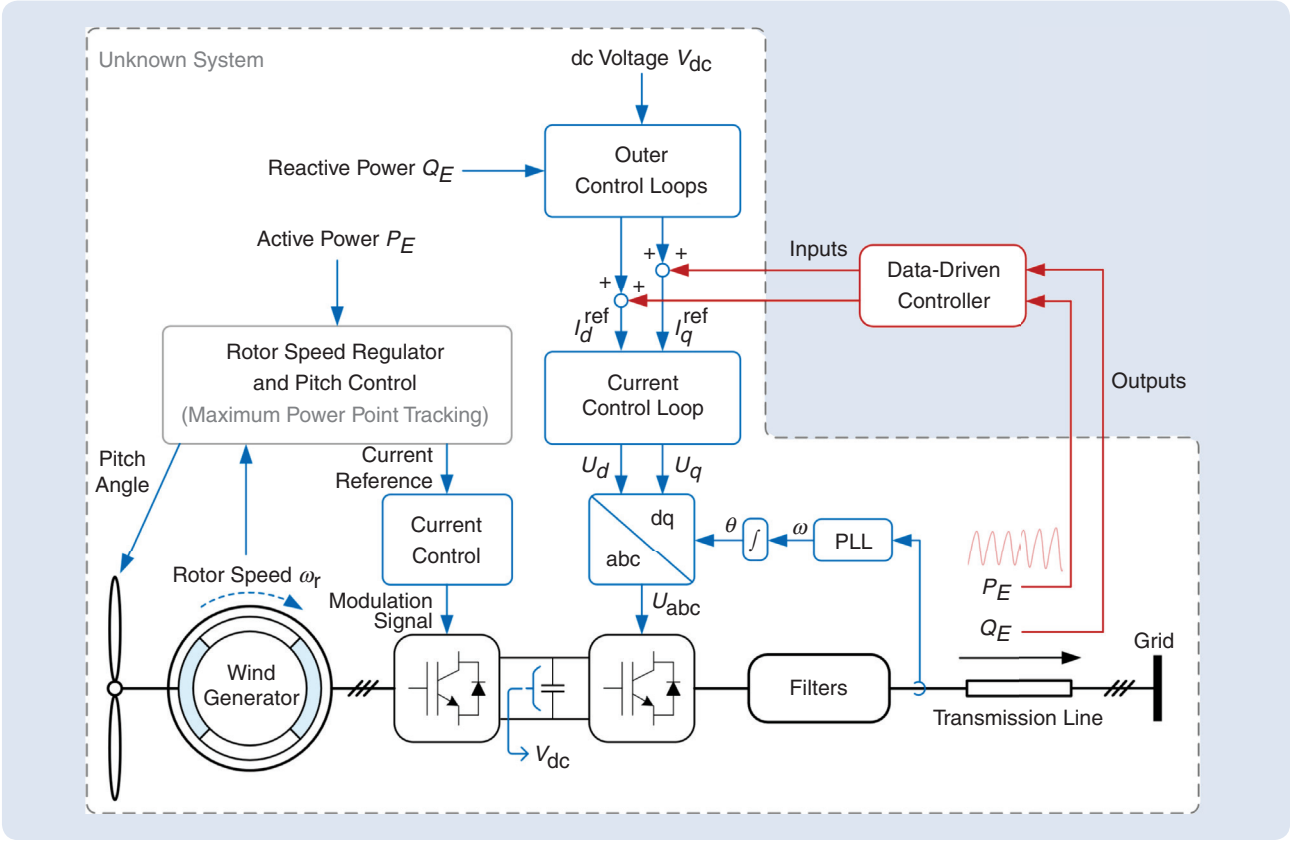



**FIGURE 11** The application of data-driven control to stabilize a direct-drive wind generator.

data that satisfy the persistency of excitation condition. The output signals are subject to measurement noise with the signal-to-noise ratio (SNR) being $2 \times 10^3$. Here, the SNR is defined as $\text{SNR} = \| \Delta y_\text{d} \|_2^2 / \| y_\text{noise} \|_2^2$, where $\Delta y_\text{d}$ is the deviation of $y_\text{d}$ from its steady-state value, and $y_\text{noise}$ is the noise sequence; it reflects how the noise compares to the fluctuation in the outputs caused by the excitation signals.

The DeePC method is activated at $time = 7$ s. It can be seen that the oscillations are well eliminated when any of the regularizers is employed. We also observe that the projection-based regularizer achieves the best damping ratio in this case. The DeePC optimization problem can be solved using standard solvers for quadratic programs. Here, we use OSQP [83] as the solver. Note that DeePC with the one-norm regularizer can also be reformulated as a quadratic program by considering an additional decision variable $g_1$ that satisfies the constraint $-g_1 \leq g \leq g_1$ such that $\| g \|_1 = \mathbf{1}^\top g_1$ in the optimizer, though this reformulation results in a higher-dimensional problem with additional constraints. On an Intel Core i7 9750H CPU with 16 GB of random-access memory, it takes about 0.013 s to solve DeePC with the quadratic regularizer, it takes about 0.02 s to solve DeePC with the one-norm regularizer, and it takes about 0.02 s to solve DeePC with the projection-based regularizer. Hence, the DeePC method can be implemented in real time by choosing a control horizon $k > 4$. When the input/output constraints are inactive, one can also use the closed-form solution to achieve faster calculations.

In what follows, we investigate how the DeePC parameters affect the closed-loop cost of the system. The closed-loop cost is defined by $\Sigma_t \| u_t \|_R^2 + \| y_t - r_t \|_Q^2$, measured from the system from 7 to 12 s. Figure 12(a) shows the average closed-loop cost of the system when different regularization coefficients $\lambda_g$ are used. The average closed-loop cost is computed by repeating the simulation 10 times with different random seeds to generate the excitation signals and the measurement noise. It can be seen that the system has a satisfactory performance for a wide range of $\lambda_g$. As mentioned before, the regularization on $g$ in the cost function ensures robustness. Hence, the regularizers achieve satisfactory performance when $\lambda_g$ is large enough.

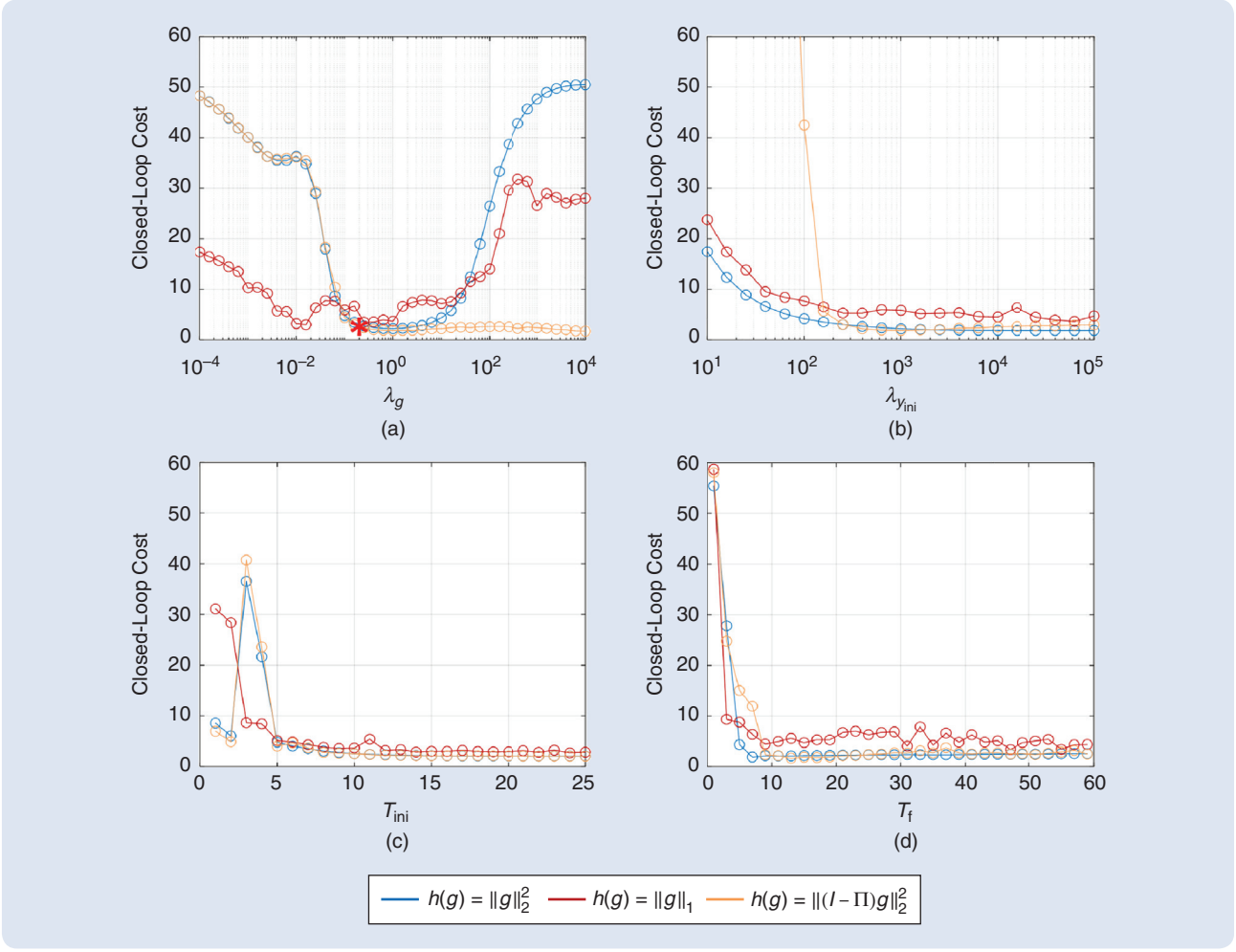



**FIGURE 12** The average realized closed-loop cost of the system (a) with different values of $\lambda_g$ (when the projection-based regularizer is used, we use $*$ to label the value of $\lambda_g$ obtained by the Hanke–Raus heuristic [81], [85]), (b) with different values of $\lambda_{y_\text{ini}}$, (c) with different values of $T_\text{ini}$, and (d) with different values of $T_\text{f}$.

However, with the quadratic regularizer or the one-norm regularizer, the performance deteriorates with a too large $\lambda_g$, as it conservatively hedges against a too large inherent uncertainty set, which results in poor performance of the obtained control sequence [64]. By comparison, the projection-based regularizer still maintains excellent performance, even with a large $\lambda_g$, as the control is robustified around the solution of the indirect problem. Moreover, we observe that the one-norm regularizer achieves better performance than the quadratic regularizer when a very low or a very high value of $\lambda_g$ is applied. In summary, Figure 12(a) quantitatively confirms the illustration in Figure 9. The Hanke–Raus heuristic [81], [85] to tune $\lambda_g$ leads to the nearly optimal value labeled by $*$ in Figure 12(a) for the projection-based regularizer. However, when the Hanke–Raus heuristic is used to tune $\lambda_g$ for the quadratic regularizer, we obtain $\lambda_g$ on the order of $10^2$, and the resulting closed-loop cost is not satisfactory.

Figure 12(b) shows the average closed-loop cost with different values of $\lambda_{y_{\text{ini}}}$, which indicates that a sufficiently large $\lambda_{y_{\text{ini}}}$ is needed to ensure satisfactory performance; see also [64, Theorem III.1]. Figure 12(c) and (d) shows the average closed-loop cost with different values of $T_{\text{ini}}$ and $T_{\text{f}}$, respectively. It can be seen that the system achieves excellent performance when $T_{\text{ini}}$ and $T_{\text{f}}$ are sufficiently large, consistent with the results in [53], [54], and [72]. Usually, the prediction horizon should be large enough to ensure the stability of the system [95]. We notice that the lag of the system $\ell$ is at least 19 since the system has 37 state variables and two outputs, and, theoretically, $T_{\text{ini}}$ should be larger than $\ell$. However, we observe from Figure 12(c) that the closed-loop system shows satisfactory performance with $T_{\text{ini}} \geq 5$. This may be because the data mainly capture the dynamics of the dominant states for the studied problem (oscillation damping). To verify our hypothesis, we use the data in the Hankel matrices to identify the system by employing structured low-rank approximation [35] with different presumed model orders. The fitting error is plotted in Figure 15, which shows that the optimal model order is $n = 5$ (in terms of balancing the fitting error and the

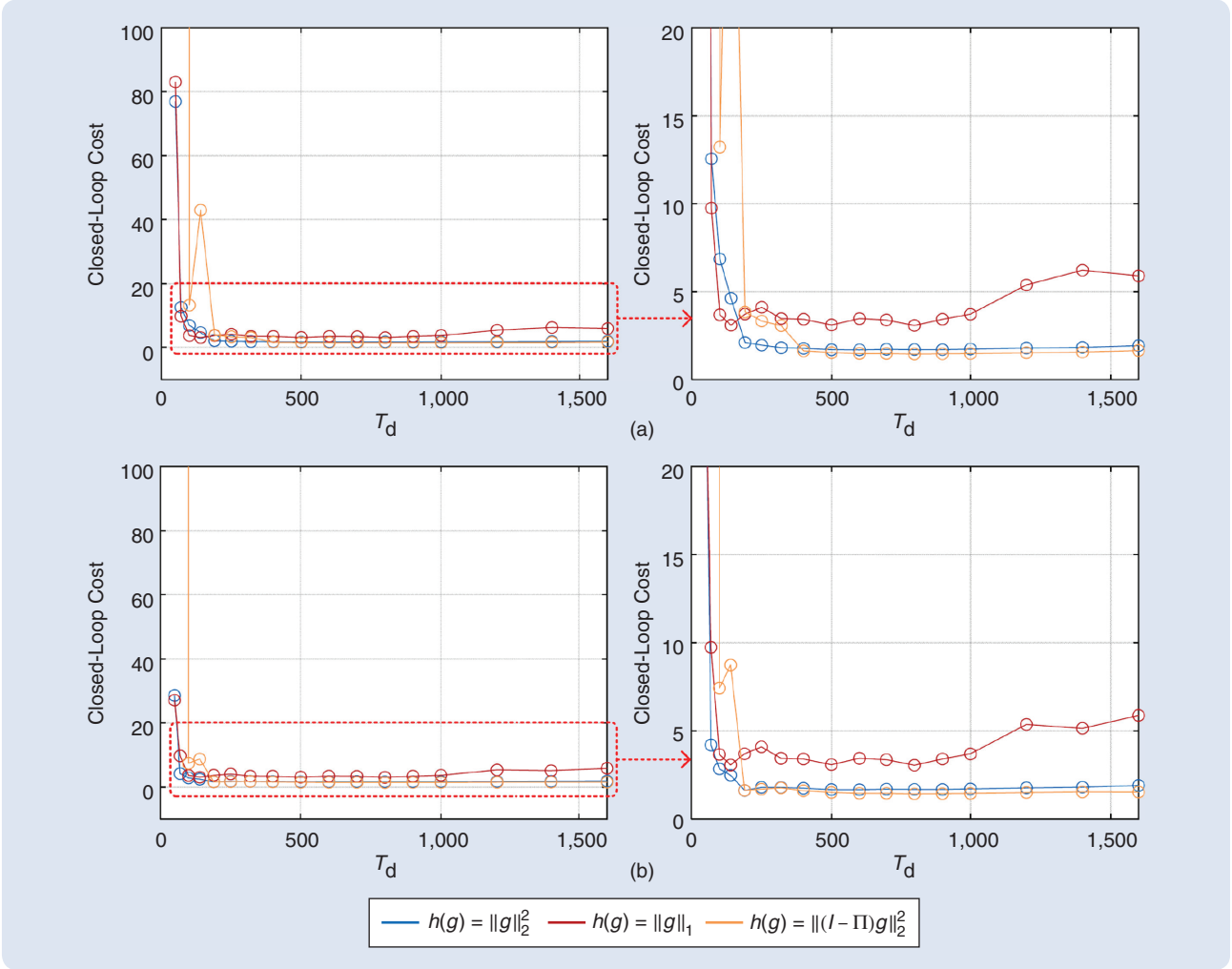



**FIGURE 13** The average realized closed-loop cost of the system with different length $T_{\text{d}}$ of input/output trajectory to construct Hankel matrices (a) with a fixed $\lambda_g$ and (b) for each value of $T_{\text{d}}$; the optimal value of $\lambda_g$ is chosen from $\{0.001, 0.01, 0.1, 1, 10, 100\}$.

complexity). Hence, aligned with insights from model-order reduction, one may expect good performance (in practice), even when $T_{\text{ini}}$ is smaller than the theoretical lower bound inferred from the real system order.

Figure 13(a) plots the average closed-loop cost with different values of $T_{\text{d}}$, which shows that a sufficiently large $T_{\text{d}}$ leads to satisfactory performance and that increasing $T_{\text{d}}$ does not always improve the performance. For instance, with the projection-based regularizer, we observe the best performance when $T_{\text{d}}$ is around 800; with the one-norm regularizer, the performance starts to degrade when $T_{\text{d}} \geq 1{,}000$. According to the interpretation from min–max optimization (See "Roles of Regularization"), one may need to choose a larger $\lambda_g$ when $T_{\text{d}}$ increases. Figure 13(b) further shows the average closed-loop cost, where, for each value of $T_{\text{d}}$, the optimal value of $\lambda_g$ (with the lowest closed-loop cost) is chosen from $\{0.001, 0.01, 0.1, 1, 10, 100\}$. The results are consistent with (and very close to) Figure 13(a). Moreover, we observe that the optimal value of $\lambda_g$ increases with a larger $T_{\text{d}}$ (data not shown). Overall, all of the regularizers have satisfactory performance when $T_{\text{d}}$ is sufficiently large.

In short, these results demonstrate the robustness of DeePC with regard to different choices of parameters. The system presents excellent control performance with deliberately tuned regularizers and sufficiently large $T_{\text{d}}$, $\lambda_{y_{\text{ini}}}$, $T_{\text{ini}}$, and $T_{\text{f}}$.

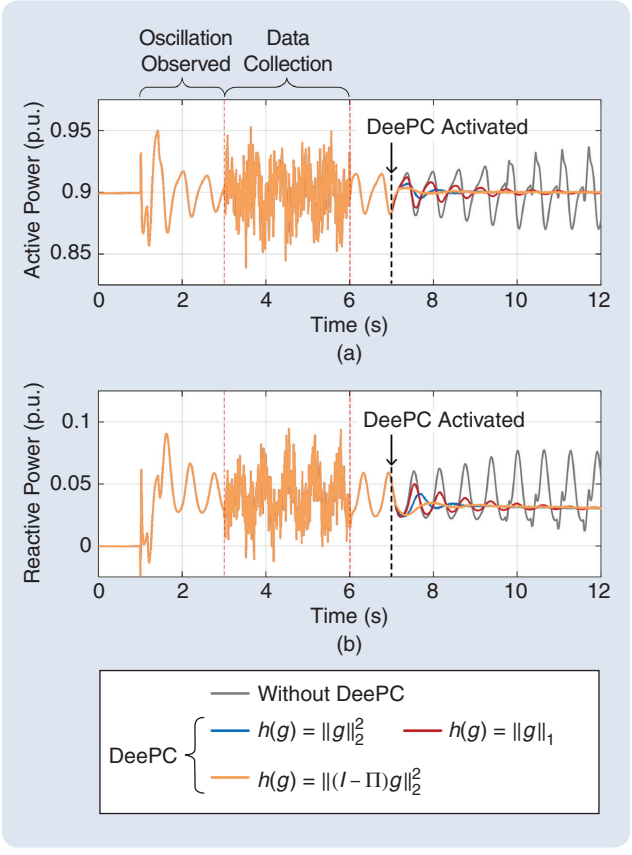



**FIGURE 14** The time-domain responses of the wind generator: (a) active power and (b) reactive power.

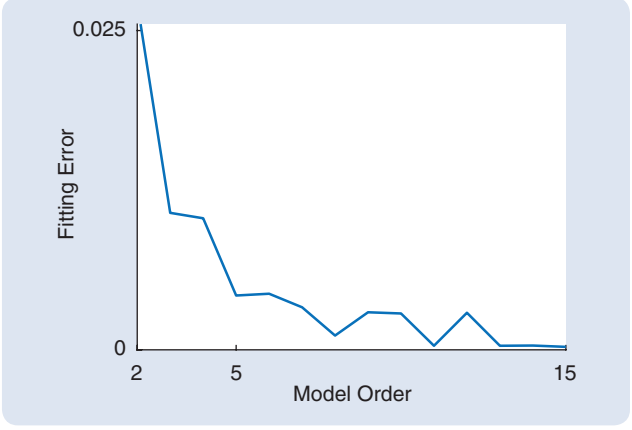



**FIGURE 15** The fitting error when assuming different model orders to identify the system.

### *Case Study 2: Grid Synchronization and DC Voltage Regulation*

In the previous case study, we used DeePC as an auxiliary control loop to stabilize wind generators, which will be

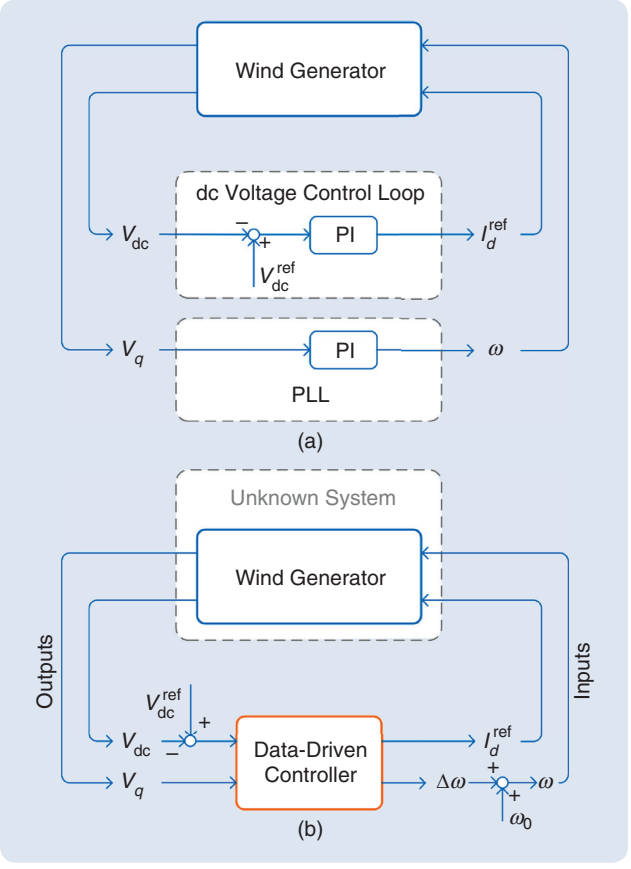



**FIGURE 16** The dc voltage control and grid synchronization control of direct-drive wind generators (a) using PI controllers and (b) using data-driven control.

activated only when oscillations/instabilities are observed. During normal operations, the wind turbines mainly use proportional-integral-derivative (PID) controllers to achieve multiple control objectives, for example, grid synchronization and dc voltage regulation. Usually, a PLL is employed to achieve grid synchronization, which uses the q-axis voltage signal $V_q$ to generate the converter's internal frequency $\omega$; see Figure 16(a). The PLL aims to regulate $V_q$ to be zero at the steady state such that the voltage vector is aligned with the d-axis. On this basis, the dc voltage control loop uses the dc voltage tracking error (that is, $\Delta V_{\rm dc} = V_{\rm dc}^{\rm ref} - V_{\rm dc}$) to generate the d-axis (active) current reference $I_d^{\rm ref}$. However, as discussed before, optimal control performance can rarely be achieved in this paradigm due to the absence of a detailed and accurate model.

In what follows, we employ DeePC to perform data-driven optimal grid synchronization control and dc voltage regulation. As shown in Figure 16(b), we use the data-driven controller to replace the conventional dc voltage control loop and PLL, which measures the system outputs $\Delta V_{\rm dc}$ and $V_q$ and provides control inputs $I_d^{\rm ref}$ and $\Delta\omega$ (the deviation of $\omega$ from the nominal frequency $\omega_0$) to the system. In this manner, the converter's frequency is generated using both the dc voltage and q-axis voltage as feedback signals. In fact, some *grid-forming converters* also use the dc voltage to generate the frequency, for example, those in [96] and [97]. Unlike PLL-based converters, which follow the grid frequency (also known as grid-following converters), grid-forming converters establish their own frequencies in the power grid and behave as coupled oscillators, which can provide fast frequency/voltage support and stabilize the power grid [98], [99], [100], [101], [102]. The data-driven controller in Figure 16(b) has the potential to become a general optimal control framework that covers both grid-forming control and PLL-based (that is, grid-following) control. An interesting question for future work is how to design the objective function of an optimal control problem to specify it as a grid-forming (or grid-following) controller.

The input/output signals are all in p.u. values, and the parameters of DeePC are as follows: $T_{\rm d} = 800$, $T_{\rm ini} = 5$, $T_{\rm f} = 25$, $R = I \otimes \mathrm{diag}(1,20)$, $Q = 1{,}000I$, and $\lambda_{y_{\rm ini}} = 1{,}000$. We choose $\lambda_g = 1$ for the quadratic regularizer, $\lambda_g = 0.01$ for the one-norm regularizer, and $\lambda_g = 2$ for the projection-based regularizer. The control horizon is $k = 1$. The reference trajectory is $r = \mathbf{1} \otimes \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]$. Note that, in the matrix $R$, we give high penalties to the future trajectory of $\Delta\omega$ to mitigate the frequency fluctuations. The sampling time is chosen as 1 ms to handle fast dc-link dynamics. In the constraints, the lower and upper bounds of $I_d^{\rm ref}$ (input 1) are –1.2 and 1.2, respectively; the lower and upper bounds of $\Delta\omega$ (input 2) are –0.01 and 0.01, respectively; the lower and upper bounds of $\Delta V_{\rm dc}$ (output 1) are –0.1 and 0.1, respectively; and the lower and upper bounds of $V_q$ (output 2) are –0.2 and 0.2, respectively. We note that, if one chooses the dc voltage $V_{\rm dc}$ as one output of the system (rather than its deviation from the reference value $\Delta V_{\rm dc}$), then the (linearized) wind generator model becomes affine. In this case, an additional constraint (S3) should be included [27].

Before DeePC is activated, we inject white noise signals for 0.8 s to excite the system and collect input/output data. During this period, $I_d^{\rm ref}$ comes from the sum of a white noise signal (variance: $4 \times 10^{-3}$) and a dc voltage control loop [shown in Figure 16(a)] that prestabilizes the dc voltage; $\omega$ comes from the sum of a white noise signal (variance: $1.75 \times 10^{-5}$) and a PLL that presynchronize the converter with the power grid during the data collection period. The output signals are subject to measurement noise (SNR: $10^4$). Under this setting, Figure 17 shows the time-domain responses of the wind generator, where the dc voltage reference steps from 1 p.u. to 1.02 p.u. at 0.5 s and steps back to 1 p.u. at 1.5 s (by changing the reference trajectory $r$ accordingly). Note that the data collection was done before $time = 0$ s, and it is not shown in Figure 17. When well-tuned PI controllers are used for the dc voltage regulation and grid synchronization, the system has large overshoots. By comparison, the DeePC method (with any regularizer) achieves excellent performance—faster responses, perfect tracking, and nearly no overshoots. We also tested the performance of applying SPC (S12), which shows faster responses but larger overshoots and tracking errors. Moreover, the responses of the q-axis voltage are smoother when DeePC is applied (with any regularizer).

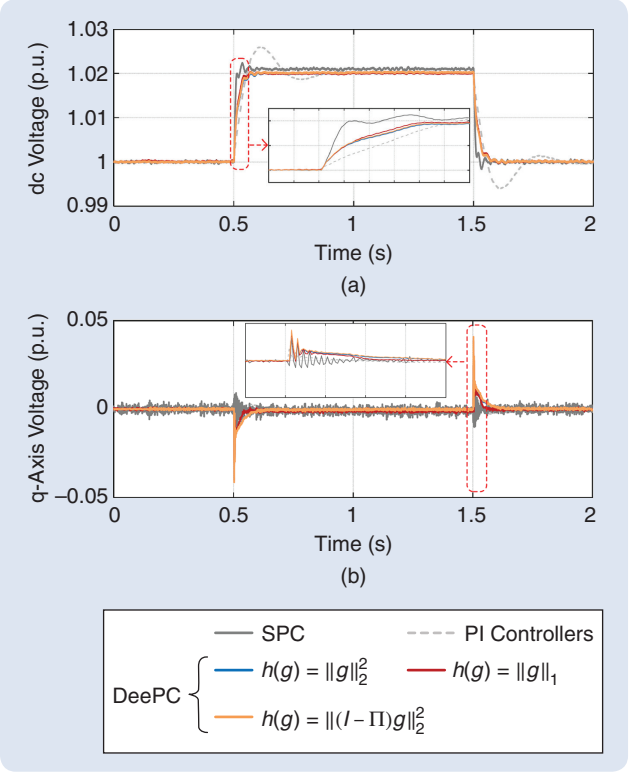



**FIGURE 17** The time-domain responses of the wind generator: (a) dc voltage and (b) q-axis voltage using PI control, SPC, and DeePC.

**TABLE 1** **Comparisons of average closed-loop costs between Hankel matrices and Page matrices.**

| Average Closed-Loop Cost | SNR: $10^4$ | SNR: $10^2$ |
|---|---|---|
| Hankel matrices ($h(g)=\|g\|_2^2$) | 17.12 | 26.69 |
| Page matrices ($h(g)=\|g\|_2^2$) | 16.5 | 24.17 |
| Hankel matrices ($h(g)=\|g\|_1$) | 31.03 | 49.04 |
| Page matrices ($h(g)=\|g\|_1$) | 24.38 | 36.76 |
| Hankel matrices ($h(g)=\|(I-\Pi)g\|_2^2$) | 31.85 | 43.12 |
| Page matrices ($h(g)=\|(I-\Pi)g\|_2^2$) | 25 | 29.21 |
| Hankel matrices (SPC) | 292.07<br>Unstable (one case) | 912.59<br>Unstable (three cases) |
| Page matrices (SPC) | $1.15\times10^3$<br>Unstable (three cases) | $2.03\times10^5$<br>Unstable (one case) |

Table 1 shows the effects of choosing Page matrices instead of Hankel matrices in DeePC. Each of the simulations is repeated 100 times with 100 different random seeds to generate the excitation signals and the measurement noise. (We use the same set of 100 random seeds across different test settings to make a fair comparison). We observe that the closed-loop performance can usually be improved by using Page matrices as predictors, which comes at the cost of collecting more input/output data. Note that we use a longer trajectory ($T_d = 23{,}130$) to construct the Page matrices such that they share the same dimensions with the Hankel matrices. Table 1 also shows the performance of SPC. However, we observe some unstable cases when SPC is applied (the unstable cases are taken into account to calculate the average costs). By comparison, all of the simulation cases are stable when DeePC is applied.

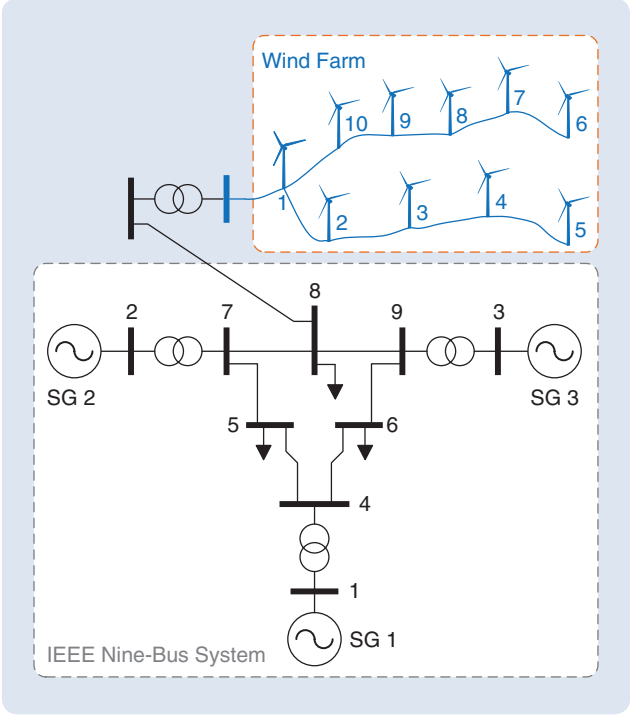



**FIGURE 18** A wind farm connected to a power grid (an IEEE nine-bus system). The wind speed at wind generators 1–5 is 10 m/s, and the wind speed at wind generators 6–10 is 9.5 m/s, due to the wake effect. The total active power output of the three SGs and the wind farm are 300 MW and 51.5 MW, respectively.

### *Case Study 3: Frequency Control Provided by Wind Farms*

In what follows, we apply DeePC to a wind farm to perform optimal frequency control and improve the frequency stability of power systems (that is, reduce frequency deviations and increase the frequency nadir). Conventionally, wind generators operate in MPPT mode and do not respond to the frequency deviation in power grids. However, this behavior will result in unacceptable frequency responses in a low-inertia system that has a high share of renewable sources [86], [103]. In fact, many system operators have required renewable sources to participate in frequency regulation. With appropriate control design in wind generators, one can use the kinetic energy stored in the spinning turbine to provide additional energy for fast frequency support [104], [105] without operating the wind generators in deloading mode. However, it is, in general, difficult to tune the control parameters when the power grid is unknown, and data-driven control can be used to handle this issue.

Consider a wind farm that is connected to a power grid (an IEEE nine-bus system), where the wind farm consists of 10 direct-drive wind generators (the same model as in the previous case studies), as shown in Figure 18. The models of the power grid and the wind farm are unknown from a control design perspective. We consider the following DeePC setting:

- The output of the unknown system is the deviation of the frequency of SG 1 from the nominal frequency (Hertz), with the lower and upper bounds being ±0.5; this output signal can be obtained from wide-area measurements.
- For each wind generator, DeePC provides an auxiliary control input signal added to the rotor speed control loop in the control scheme of the generator-side converter (see Figure 19), with the lower and upper bounds being ±0.3.
- The parameters of DeePC are as follows: $T_d = 500$, $T_{ini} = 10$, $T_f = 25$, $R = I$, $Q = 40I$, $\lambda_{y_{ini}} = 40$, and $k = 1$; we choose $\lambda_g = 2$ for the quadratic regularizer and $\lambda_g = 0.4$ for the one-norm regularizer and the projection-based regularizer.

» We choose the sampling time to be 0.1 s, as we focus on slow power grid frequency dynamics.

During the data collection period, we simultaneously inject the same excitation sequence into all of the wind generators to cause fluctuations in the system output (within 0.05 Hz); the output is subject to measurement noise (SNR: $5 \times 10^5$). Under this setting, since every wind generator collects the same input/output data, one can implement DeePC in one central controller using wide-area measurements and then distribute the obtained control sequence to all of the wind generators; that is, they operate together as a virtual power plant [106]. The implementation of this setting is illustrated in Figure 19. DeePC can also be implemented locally in

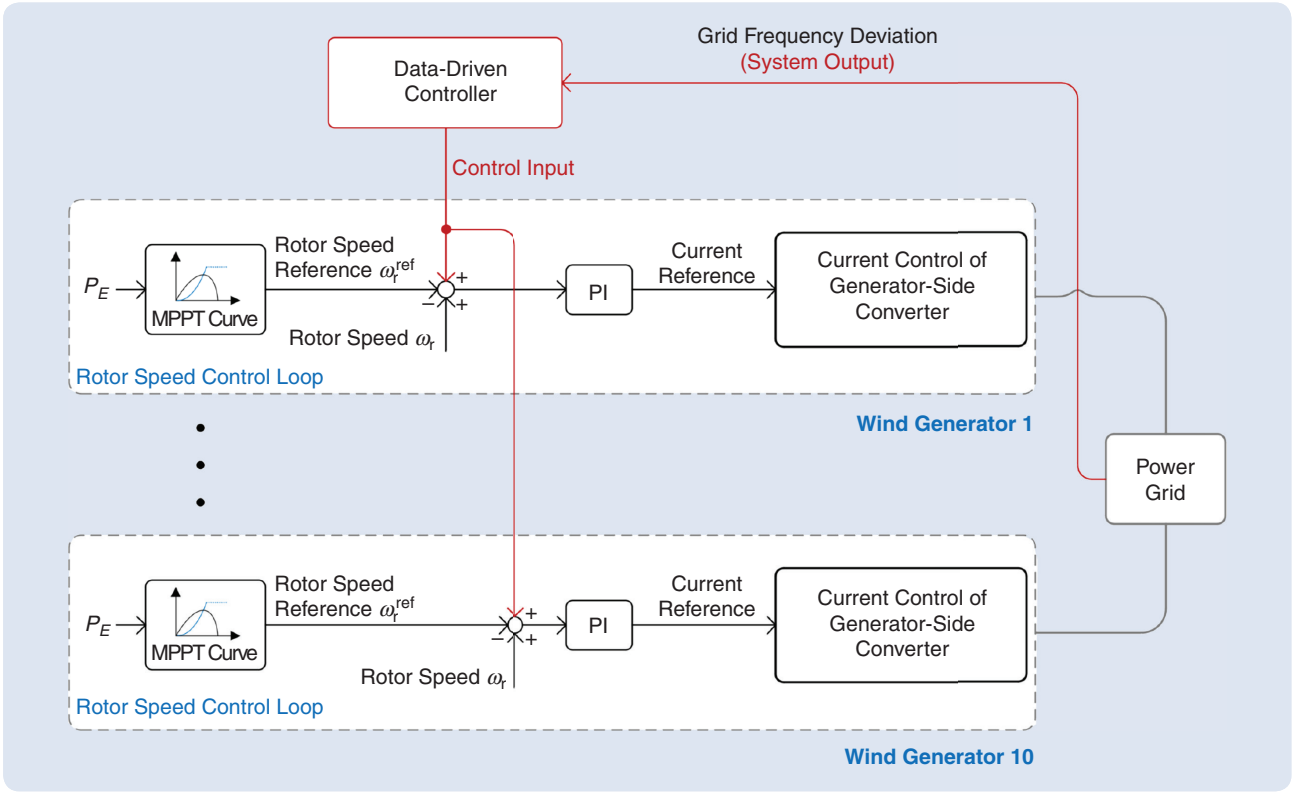



**FIGURE 19** A centralized implementation of data-driven frequency control in the rotor speed control loop.

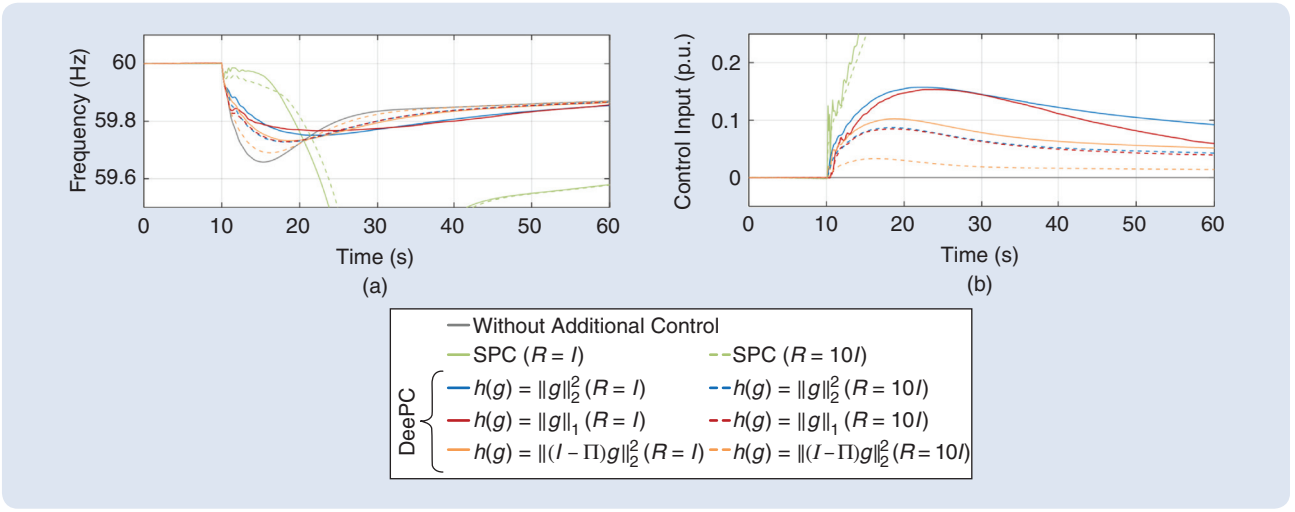



**FIGURE 20** The time-domain responses of the system: (a) the frequency of SG 1 and (b) the control inputs provided by DeePC. The wind generators release kinetic energy to support the grid frequency without operating in deloaded mode, which significantly increases the frequency nadir. At the same time, the frequency recovery process becomes slower as the wind generators need to absorb energy to restore to the original speed (see also Figure 21).

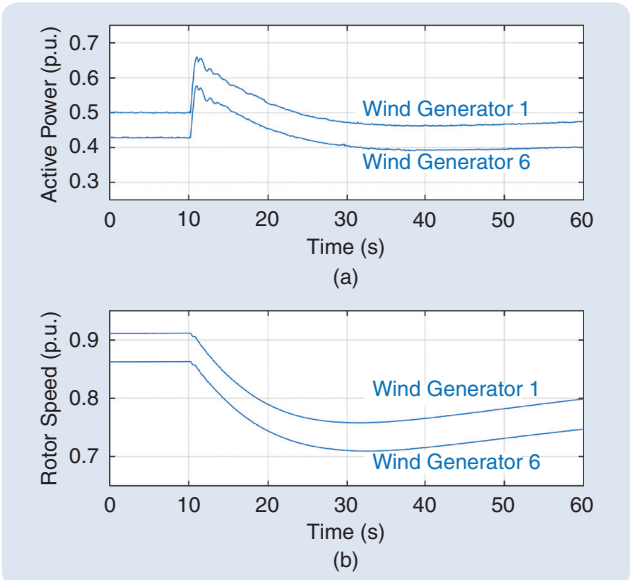



**FIGURE 21** The time-domain responses of the wind generators with DeePC ($h(g) = \|g\|_2^2, R = I$): (a) active power and (b) rotor speed. The wind generators decelerate to provide kinetic energy to support the grid frequency.

each wind generator to avoid delays in the control actuation; see [72] for an example. We later present a decentralized implementation using PLLs to measure the grid frequency locally. In practice, one can possibly avoid actively perturbing the system by collecting long enough operation data of the wind farm, as the persistency of excitation condition can generally be met thanks to the fluctuation of wind power.

Figure 20 shows the time-domain responses of the system when an overload event (bus 6) occurs at $time = 10$ s and causes a frequency dip. The data collection was done before $time = 0$ s, and it is not shown in Figure 20. We employ DeePC in a central controller that provides an identical control sequence to all of the wind generators. It can be seen from Figure 20(a) that the frequency nadir is improved when DeePC is applied, indicating better frequency stability. For instance, the frequency nadir is lifted from 59.66 to 59.75 Hz when the quadratic regularizer is used ($R = I$). In our case, the power generation of the wind farm is only 15% of the total generation in the grid, and the frequency nadir can be improved more if the wind power has a larger share. Moreover, the performance improvement does not come at

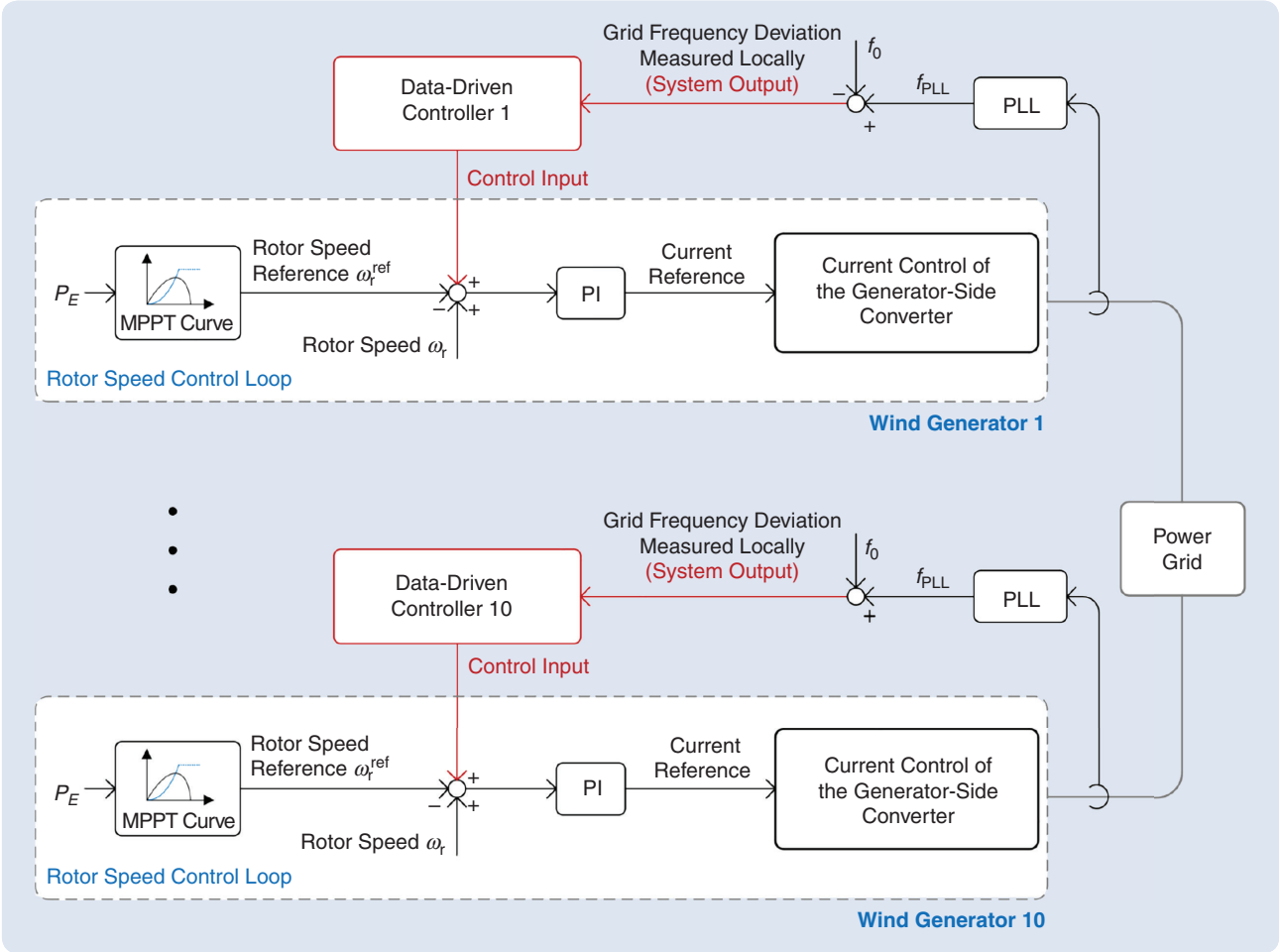



**FIGURE 22** A fully decentralized implementation of data-driven frequency control in the rotor speed control loops of wind generators, where $f_{PLL}$ is the PLL frequency, and $f_0$ is the nominal frequency.

the cost of operating in deloading mode, as we utilize the kinetic energy stored in the spinning turbine to support the grid frequency. Figure 20 further compares the responses with different values of $R$, which shows that the control action is mitigated with a larger $R$. Hence, one can conveniently reduce the frequency support capability of the wind generators by choosing a larger $R$ that penalizes more on the control effort. We also test the performance of SPC. However, as shown in Figure 20, the system becomes unstable in this case. We suspect that this is due to the nonlinearity of the system (for example, the nonlinear MPPT curve and the nonlinear relationship between the captured wind power and rotor speed).

Figure 21 shows the active power and rotor speed of the wind generators when the quadratic regularizer is used ($\lambda_g = 2,\ R = I$). Note that wind generators 2–5 have similar responses to wind generator 1, and wind generators 7–10 have similar responses to wind generator 6. It can be seen that the wind generators decelerate to release the kinetic energy, and, thus, the active power increases quickly after the overload event. Note that the wind generators deviate from the MPPT curve during this process, and the rotor speeds will be restored to the original point to preserve the MPPT operation once the secondary frequency control of the power grid brings the frequency back to the nominal value. (This process is not shown, as it usually takes several minutes.) In these simulations, it only takes about 0.02 s to solve the optimization problem, and, thus, the algorithm can be implemented in real time.

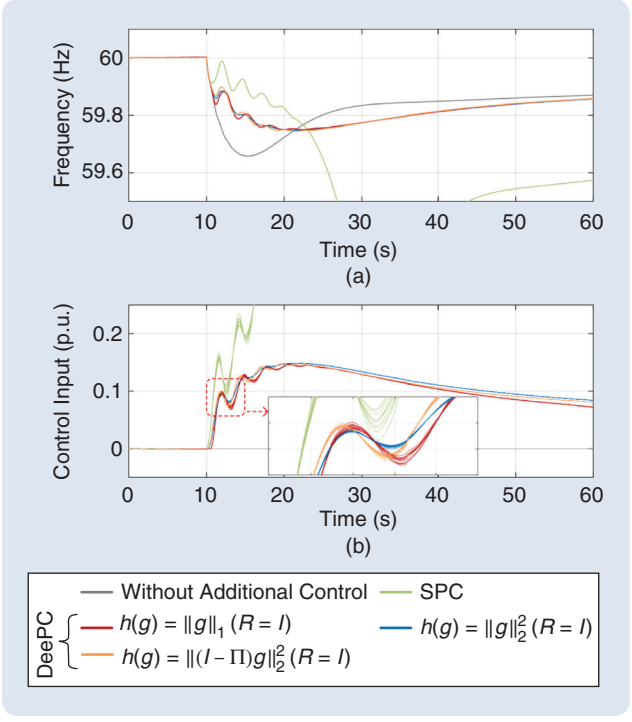



**FIGURE 23** The time-domain responses when PLL measurements are used: (a) the frequency of SG 1 and (b) the control inputs of all of the wind generators provided by DeePC in a decentralized manner.

Instead of directly using wide-area measurements, we next consider the scenario where each wind generator uses an additional PLL (bandwidth: 5 rad/s) to locally detect the grid frequency and implement DeePC in a decentralized manner, as shown in Figure 22. Again, we simultaneously inject the same excitation sequence into all of the wind generators during the data collection period. However, different wind generators will observe different grid frequency responses through their own PLLs. Figure 23 displays the responses of the system under this setting, which shows that DeePC provides similar control inputs to the wind generators when implemented in a decentralized manner. Again, DeePC (with any regularizer) effectively improves the frequency nadir. We choose $\lambda_g = 2$ for the quadratic regularizer, $\lambda_g = 0.4$ for the one-norm regularizer, and $\lambda_g = 0.1$ for the projection-based regularizer. Under this setting, the three different regularizers have similar responses and performance, which significantly increase the frequency nadir and, thus, improve the frequency stability. Note that the DeePC hyperparameters are not optimized in our case studies, and one can achieve better performance by optimizing over the parameters. We again observe that the system becomes unstable when applying SPC in this case. Compared with Figure 20, the frequency responses in Figure 23 have lower but still satisfactory damping ratio. Overall, these results demonstrate the great potential of applying DeePC to perform optimal frequency regulation in power systems.

### *Summary*

This section has presented the applications of DeePC in power systems, including 1) the stabilization of wind generators, 2) grid synchronization and dc voltage regulation, and 3) frequency control provided by wind farms. Our results suggest a great potential for data-driven control in handling unknown and highly complex power grid dynamics as well as ensuring a resilient power electronics-dominated power system.

## CONCLUSION

The data-driven representation of the finite-horizon behavior made possible the development of novel methods for analysis, signal processing, and control that achieve a direct map from observed data of the system to a desired solution. For exact noise-free data of LTI systems, the methods require simple linear algebra operations and have no hyperparameters. Modification using regularization techniques makes the methods effective for noisy data and nonlinear dynamics.

Current limitations of the direct data-driven approach presented in the article are as follows: data and computational efficiency for growing prediction horizons, lack of statistical analysis/guarantees in the case of noisy data, and the direct design of feedback controllers. These limitations open

fruitful avenues for future research, including a principled and bottom-up extension of the methods to continuous-time, stochastic, and nonlinear systems starting from their behaviors. We believe that the real-time adaptation of the presented methods holds great value both for theory and for practical implementations. For the latter, we also highlight automatic tuning of hyperparameters as well as computationally efficient methods as important directions.

## ACKNOWLEDGMENT

The authors wish to thank Hamid Ossareh, Jeremy Coulson, John Lygeros, Panos Patrinos, and Leander Hemelhof for many discussions leading up to and improving this survey article.

The research leading to these results received funding from the Catalan Institution for Research and Advanced Studies; Fund for Scientific Research–Vlaanderen (FWO) Projects G090117N and G033822N; FNRS–FWO EOS Project 30468160; European Union's Horizon 2020 research and innovation program (Grant Agreement 883985); Swiss National Science Foundation under Project 200021E_203979; and NCCR Automation and ETH Zurich Funds. Ivan Markovsky and Linbin Huang are co-first authors.

## AUTHOR INFORMATION

***Ivan Markovsky*** (imarkovsky@cimne.upc.edu) is a Catalan Institution for Research and Advanced Studies professor at the International Centre for Numerical Methods in Engineering, 23 Barcelona, Spain. He received the Ph.D. degree in electrical engineering from the Katholieke Universiteit Leuven in February 2005. From 2006 to 2012, he was an assistant professor at the School of Electronics and Computer Science of the University of Southampton and from 2012 to 2022 an associate professor at the Vrije Universiteit Brussel. He is a recipient of a European Research Council starting grant "Structured Low-Rank Approximation: Theory, Algorithms, and Applications" in 2010–2015, Householder Prize honorable mention in 2008, and research mandate by the Vrije Universiteit Brussel research council in 2012–2022. His research interests include computational methods for system theory, identification, and data-driven control in the behavioral setting.

***Linbin Huang*** is a postdoctoral researcher with the Automatic Control Laboratory at ETH Zürich, 8092 Zürich, Switzerland. He received the B.Eng. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2015 and 2020, respectively. His research interests include power system stability, optimal control of power electronics, and data-driven control.

***Florian Dörfler*** is an associate professor at the Automatic Control Laboratory at ETH Zürich, 8092 Zürich, Switzerland, and the associate head of the Department of Information Technology and Electrical Engineering. He received the Ph.D. degree in mechanical engineering from the University of California, Santa Barbara in 2013 and the Diplom degree in engineering cybernetics from the University of Stuttgart in 2008. From 2013 to 2014, he was an assistant professor at the University of California, Los Angeles. His research interests include control, optimization, and system theory with applications in network systems, in particular, electric power grids. He is a recipient of the Distinguished Young Research Awards by the International Federation of Automatic Control (Manfred Thoma Medal, 2020) and European Control Association (European Control Award, 2020). His students were winners or finalists for Best Student Paper Awards at the European Control Conference, 2013 and 2019; the American Control Conference, 2016; the Conference on Decision and Control, 2020; the IEEE Power & Energy Society (PES) General Meeting, 2020; and the PES PowerTech Conference, 2017. He is, furthermore, a recipient of the 2010 American Control Conference Student Best Paper Award, the 2011 O. Hugo Schuck Best Paper Award, the 2012–2014 Automatica Best Paper Award, the 2016 IEEE Circuits and Systems Guillemin–Cauer Best Paper Award, and the 2015 University of California Santa Barbara Mechanical Engineering Best Ph.D. Award.

## REFERENCES

[1] K. Popper, *Conjectures and Refutations: The Growth of Scientific Knowledge*. Evanston, IL, USA: Routledge, 1963.

[2] L. Ljung, "Identification for control: Simple process models," in *Proc. 41st IEEE Conf. Decis. Contr.*, Dec. 2002, vol. 4, pp. 4652–4657, doi: 10.1109/CDC.2002.1185112.

[3] M. Gevers, "Identification for control: From the early achievements to the revival of experiment design," *Eur. J. Contr.*, vol. 11, no. 4, pp. 335–352, 2005, doi: 10.3166/ejc.11.335-352.

[4] H. Hjalmarsson, "From experiment design to closed-loop control," *Automatica*, vol. 41, no. 3, pp. 393–438, Jun. 2005, doi: 10.1016/j.automatica.2004.11.021.

[5] A. Feldbaum, "Dual control theory problems," *IFAC Proc. Vol.*, vol. 1, no. 2, pp. 541–550, Jun. 1963, doi: 10.1016/S1474-6670(17)69687-3.

[6] M. Ferizbegovic, J. Umenberger, H. Hjalmarsson, and T. B. Schön, "Learning robust LQ-controllers using application oriented exploration," *IEEE Contr. Syst. Lett.*, vol. 4, no. 1, pp. 19–24, Jan. 2020, doi: 10.1109/LCSYS.2019.2921512.

[7] C. A. Larsson, A. Ebadat, C. R. Rojas, X. Bombois, and H. Hjalmarsson, "An application-oriented approach to dual control with excitation for closed-loop identification," *Eur. J. Contr.*, vol. 29, pp. 1–16, May 2016, doi: 10.1016/j.ejcon.2016.03.001.

[8] A. Iannelli, M. Khosravi, and R. Smith, "Structured exploration in the finite horizon linear quadratic dual control problem," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 959–964, Apr. 2021, doi: 10.1016/j.ifacol.2020.12.1263.

[9] A. Chiuso and G. Pillonetto, "System identification: A machine learning perspective," *Annu. Rev. Contr., Robot., Auton. Syst.*, vol. 2, pp. 281–304, May 2019, doi: 10.1146/annurev-control-053018-023744.

[10] S. Formentin and A. Chiuso, "Control-oriented regularization for linear system identification," *Automatica*, vol. 127, Mar. 2021, Art. no. 109539, doi: 10.1016/j.automatica.2021.109539.

[11] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct and indirect data-driven control formulations via regularizations and relaxations," *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 883–897, Feb. 2023, doi: 10.1109/TAC.2022.3148374.

[12] P. Stoica and Y. Selén, "Model-order selection: A review of information criterion rules," *IEEE Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, Jul. 2004, doi: 10.1109/MSP.2004.1311138.

[13] J. C. Willems, "From time series to linear system—Part I. Finite dimensional linear time invariant systems," *Automatica*, vol. 22, no. 5, pp. 561–580, Sep. 1986, doi: 10.1016/0005-1098(86)90066-X

[14] J. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory*. New York, NY, USA: Springer-Verlag, 1998.

[15] J. C. Willems, "The behavioral approach to open and interconnected systems: Modeling by tearing, zooming, and linking," *IEEE Control Syst. Mag.*, vol. 27, no. 6, pp. 46–99, Dec. 2007, doi: 10.1109/MCS.2007.906923.

[16] J. C. Willems, "Terminals and ports," *IEEE Circuits Syst. Mag.*, vol. 10, no. 4, pp. 8–26, Fourthquarter 2010, doi: 10.1109/MCAS.2010.938635.
[17] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annu. Rev. Contr.*, vol. 52, pp. 42–64, Dec. 2021, doi: 10.1016/j.arcontrol.2021.09.005.
[18] K. Usevich and I. Markovsky, "Variable projection for affinely structured low-rank approximation in weighted 2-norms," *J. Comput. Appl. Math.*, vol. 272, pp. 430–448, Dec. 2014, doi: 10.1016/j.cam.2013.04.034.
[19] A. Damen, P. Van den Hof, and A. Hajdasinski, "Approximate realization based upon an alternative to the Hankel matrix: The page matrix," *Syst. Contr. Lett.*, vol. 2, no. 4, pp. 202–208, Dec. 1982, doi: 10.1016/0167-6911(82)90002-0.
[20] S. Brunton, J. Proctor, and N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Acad. Sci.*, vol. 113, pp. 3932–3937, Apr. 2016, doi: 10.1073/pnas.1517384113.
[21] J. C. Willems, P. Rapisarda, I. Markovsky, and B. De Moor, "A note on persistency of excitation," *Syst. Contr. Lett.*, vol. 54, no. 4, pp. 325–329, Apr. 2005, doi: 10.1016/j.sysconle.2004.09.003.
[22] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Boston, MA, USA: Kluwer, 1996.
[23] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *Int. J. Contr.*, vol. 81, no. 12, pp. 1946–1959, Oct. 2008, doi: 10.1080/00207170801942170.
[24] I. Markovsky, J. C. Willems, P. Rapisarda, and B. De Moor, "Algorithms for deterministic balanced subspace identification," *Automatica*, vol. 41, no. 5, pp. 755–766, May 2005, doi: 10.1016/j.automatica.2004.10.007.
[25] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Trans. Autom. Control*, vol. 67, pp. 3289–3304, Jul. 2022, doi: 10.1109/TAC.2021.3097706.
[26] V. Mishra, I. Markovsky, and B. Grossmann, "Data-driven tests for controllability," *IEEE Contr. Syst. Lett.*, vol. 5, no. 2, pp. 517–522, Apr. 2021, doi: 10.1109/LCSYS.2020.3003770.
[27] J. Berberich, J. Köhler, M. A. Muller, and F. Allgower, "Linear tracking MPC for nonlinear systems—Part II: The data-driven case," *IEEE Trans. Autom. Control*, vol. 67, no. 9, pp. 4406–4421, Sep. 2022, doi: 10.1109/TAC.2022.3166851.
[28] B. L. Ho and R. E. Kalman, "Effective construction of linear state-variable models from input/output functions," *Regelungstechnik*, vol. 14, no. 12, pp. 545–592, Dec. 1966, doi: 10.1524/auto.1966.14.112.545.
[29] I. Markovsky, *Low-Rank Approximation: Algorithms, Implementation, Applications*, 2nd ed. Cham, Switzerland: Springer, 2019.
[30] I. Markovsky and K. Usevich, "Structured low-rank approximation with missing data," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 2, pp. 814–830, 2013, doi: 10.1137/120883050.
[31] I. Markovsky and F. Dörfler, "Data-driven dynamic interpolation and approximation," *Automatica*, vol. 135, Jan. 2022, Art. no. 110008, doi: 10.1016/j.automatica.2021.110008.
[32] I. Markovsky and F. Dörfler, "Identifiability in the behavioral setting," *IEEE Trans. Autom. Control*, vol. 68, no. 3, pp. 1667–1677, Mar. 2023, doi: 10.1109/TAC.2022.3209954.
[33] T. Söderström, "Errors-in-variables methods in system identification," *Automatica*, vol. 43, no. 6, pp. 939–958, Jun. 2007, doi: 10.1016/j.automatica.2006.11.025.
[34] I. Markovsky and K. Usevich, "Software for weighted structured low-rank approximation," *J. Comput. Appl. Math.*, vol. 256, pp. 278–292, Jan. 2014, doi: 10.1016/j.cam.2013.07.048.
[35] I. Markovsky, "A software package for system identification in the behavioral setting," *Contr. Eng. Pract.*, vol. 21, no. 10, pp. 1422–1436, Oct. 2013, doi: 10.1016/j.conengprac.2013.06.010.
[36] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2002.
[37] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel, "DAISY: A database for identification of systems," *J. A*, vol. 38, no. 3, pp. 4–5, Oct. 1997.
[38] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. New York, NY, USA: Dover, 2008.
[39] F. Dörfler, P. Tesi, and C. de Persis, "On the role of regularization in direct data-driven LQR control," in *Proc. 61st IEEE Conf. Decis. Contr.*, Dec. 2022, pp. 1091–1098, doi: 10.1109/CDC51059.2022.9992770.
[40] L. Campestrini, D. Eckhard, A. S. Bazanella, and M. Gevers, "Data-driven model reference control design by prediction error identification," *J. Franklin Inst.*, vol. 354, no. 6, pp. 2628–2647, Apr. 2017, doi: 10.1016/j.jfranklin.2016.08.006.
[41] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 909–924, Mar. 2020, doi: 10.1109/TAC.2019.2959924.
[42] H. van Waarde, J. Eising, H. Trentelman, and K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Trans. Autom. Control*, vol. 65, no. 11, pp. 4753–4768, Nov. 2020, doi: 10.1109/TAC.2020.2966717.
[43] J. Berberich, C. W. Scherer, and F. Allgöwer, "Combining prior knowledge and data for robust controller design," 2020, *arXiv:2009.05253*.
[44] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, vol. 2. Madison, WI, USA: Nob Hill Publishing, 2017.
[45] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[46] J. Willems, "In control, almost from the beginning until the day after tomorrow," *Eur. J. Contr.*, vol. 13, no. 1, pp. 71–81, Jan. 2007, doi: 10.3166/ejc.13.71-81.
[47] B. Anderson and J. Moore, *Optimal Control: Linear Quadratic Methods*. Chelmsford, MA, USA: Courier Corporation, 2007.
[48] J. Köhler, K. Wabersich, J. Berberich, and M. Zeilinger, "State space models vs. multi-step predictors in predictive control: Are state space models complicating safe data-driven designs?" in *Proc. IEEE Conf. Decis. Contr.*, 2022, pp. 491–498.
[49] W. Favoreel, B. De Moor, and M. Gevers, "SPC: Subspace predictive control," *IFAC Proc. Vol.*, vol. 32, no. 2, pp. 4004–4009, Jul. 1999, doi: 10.1016/S1474-6670(17)56683-5.
[50] B. Huang and R. Kadali, *Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-Driven Subspace Approach*. London, U.K.: Springer-Verlag, 2008.
[51] E. Frazzoli, M. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1077–1091, Dec. 2005, doi: 10.1109/TRO.2005.852260.
[52] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives," in *Proc. Amer. Contr. Conf.*, 2012, pp. 4239–4244, doi: 10.1109/ACC.2012.6315303.
[53] E. Elokda, J. Coulson, P. N. Beuchat, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for quadcopters," *Int. J. Robust Nonlinear Contr.*, vol. 31, no. 18, pp. 8916–8936, Jul. 2021, doi: 10.1002/rnc.5686.
[54] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for grid-connected power converters," in *Proc. IEEE Conf. Decis. Contr.*, 2019, pp. 8130–8135, doi: 10.1109/CDC40024.2019.9029522.
[55] F. Fiedler and S. Lucia, "On the relationship between data-enabled predictive control and subspace predictive control," in *Proc. Eur. Contr. Conf.*, 2021, pp. 222–229, doi: 10.23919/ECC54610.2021.9654975.
[56] V. Breschi, A. Chiuso, and S. Formentin, "Data-driven predictive control in a stochastic setting: A unified framework," 2022, *arXiv:2203.10846*.
[57] C. Cutler and B. Ramaker, "Dynamic matrix control – A computer control algorithm," in *Proc. Joint Autom. Contr. Conf.*, 1980, no. 17, p. 72.
[58] C. Garcia, D. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989, doi: 10.1016/0005-1098(89)90002-2.
[59] L. Hewing, K. Wabersich, M. Menner, and M. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Contr., Robot., Auton. Syst.*, vol. 3, pp. 269–296, May 2020, doi: 10.1146/annurev-control-090419-075625.
[60] A. Mesbah et al., "Fusion of machine learning and MPC under uncertainty: What advances are on the horizon?" in *Proc. Amer. Contr. Conf.*, 2022, pp. 342–357, doi: 10.23919/ACC53348.2022.9867643.
[61] L. Brunke et al., "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annu. Rev. Contr., Robot., Auton. Syst.*, vol. 5, pp. 411–444, May 2022, doi: 10.1146/annurev-control-042920-020211.
[62] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *Proc. Eur. Contr. Conf.*, 2019, pp. 307–312, doi: 10.23919/ECC.2019.8795639.
[63] F. Dörfler, P. Tesi, and C. De Persis, "On the certainty-equivalence approach to direct data-driven LQR design," 2021, *arXiv:2109.06643*.
[64] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Robust data-enabled predictive control: Tractable formulations and performance guarantees," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 3163–3170, May 2023, doi: 10.1109/TAC.2023.3241282.
[65] V. Krishnan and F. Pasqualetti, "On direct vs indirect data-driven predictive control," in *Proc. 60th IEEE Conf. Decis. Contr.*, 2021, pp. 736–741, doi: 10.1109/CDC45484.2021.9683187.
[66] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Quadratic regularization of data-enabled predictive control: Theory and application to power

converter experiments," *IFAC-PapersOnLine*, vol. 54, no. 7, Sep. 2021, pp. 192–197, doi: 10.1016/j.ifacol.2021.08.357.
[67] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021, doi: 10.1109/TAC.2020.3000182.
[68] A. Xue and N. Matni, "Data-driven system level synthesis," in *Proc. 3rd Conf. Learn. Dyn. Contr.*, PMLR, 2021, pp. 189–200.
[69] L. Furieri, B. Guo, A. Martin, and G. Ferrari-Trecate, "Near-optimal design of safe output-feedback controllers from noisy data," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2699–2714, May 2023, doi: 10.1109/TAC.2022.3180692.
[70] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, "System level synthesis," *Annu. Rev. Contr.*, vol. 47, pp. 364–393, Jun. 2019, doi: 10.1016/j.arcontrol.2019.03.006.
[71] L. Furieri, Y. Zheng, A. Papachristodoulou, and M. Kamgarpour, "An input–output parametrization of stabilizing controllers: Amidst Youla and system level synthesis," *IEEE Control Syst. Lett.*, vol. 3, no. 4, pp. 1014–1019, Oct. 2019, doi: 10.1109/LCSYS.2019.2920205.
[72] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Decentralized data-enabled predictive control for power system oscillation damping," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 3, pp. 1065–1077, May 2022, doi: 10.1109/TCST.2021.3088638.
[73] M. Yin, A. Iannelli, and R. S. Smith, "Maximum likelihood estimation in data-driven modeling and control," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 317–328, Jan. 2023, doi: 10.1109/TAC.2021.3137788.
[74] J. Coulson, J. Lygeros, and F. Dörfler, "Regularized and distributionally robust data-enabled predictive control," in *Proc. IEEE Conf. Decis. Contr.*, 2019, pp. 7165–7170, doi: 10.1109/CDC40024.2019.9028943.
[75] Y. Lian, J. Shi, M. P. Koch, and C. N. Jones, "Adaptive robust data-driven building control via bi-level reformulation: An experimental result," 2021, *arXiv:2106.05740*.
[76] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven tracking MPC for changing setpoints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6923–6930, Apr. 2021, doi: 10.1016/j.ifacol.2020.12.389.
[77] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "On the design of terminal ingredients for data-driven MPC," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 257–263, Sep. 2021, doi: 10.1016/j.ifacol.2021.08.554.
[78] J. Bongard, J. Berberich, J. Köhler, and F. Allgower, "Robust stability analysis of a simple data-driven model predictive control approach," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2625–2637, May 2023, doi: 10.1109/TAC.2022.3163110.
[79] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Robust constraint satisfaction in data-driven MPC," in *Proc. 59th IEEE Conf. Decis. Contr.*, 2020, pp. 1260–1267, doi: 10.1109/CDC42340.2020.9303965.
[80] J. Berberich, J. Köhler, M. Müller, and F. Allgöwer, "Stability in data-driven MPC: An inherent robustness perspective," in *Proc. 61st IEEE Conf. Decis. Contr.*, Dec. 2022, pp. 1105–1110, doi: 10.1109/CDC51059.2022.9993361.
[81] M. Lazar and P. Verheijen, "Offset-free data-driven predictive control," in *Proc. 61st IEEE Conf. Decis. Contr.*, Dec. 2022, pp. 1099–1104, doi: 10.1109/CDC51059.2022.9992453.
[82] D. Alpago, F. Dörfler, and J. Lygeros, "An extended Kalman filter for data-enabled predictive control," *IEEE Contr. Syst. Lett.*, vol. 4, no. 4, pp. 994–999, Oct. 2020, doi: 10.1109/LCSYS.2020.2998296.
[83] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Math. Program. Comput.*, vol. 12, no. 4, pp. 637–672, Feb. 2020, doi: 10.1007/s12532-020-00179-2.
[84] R. Ou, G. Pan, and T. Faulwasser, "Data-driven multiple shooting for stochastic optimal control," *IEEE Contr. Syst. Lett.*, vol. 7, pp. 313–318, 2023, doi: 10.1109/LCSYS.2022.3185841.
[85] M. Hanke and T. Raus, "A general heuristic for choosing the regularization parameter in ill-posed problems," *SIAM J. Scientific Comput.*, vol. 17, no. 4, pp. 956–972, 1996, doi: 10.1137/0917062.
[86] F. Milano, F. Dörfler, G. Hug, D. J. Hill, and G. Verbič, "Foundations and challenges of low-inertia systems," in *Proc. Power Syst. Comput. Conf.*, 2018, pp. 1–25, doi: 10.23919/PSCC.2018.8450880.
[87] B. Kroposki et al., "Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy," *IEEE Power Energy Mag.*, vol. 15, no. 2, pp. 61–73, Mar./Apr. 2017, doi: 10.1109/MPE.2016.2637122.
[88] F. Blaabjerg, Z. Chen, and S. B. Kjaer, "Power electronics as efficient interface in dispersed power generation systems," *IEEE Trans. Power Electron.*, vol. 19, no. 5, pp. 1184–1194, Sep. 2004, doi: 10.1109/TPEL.2004.833453.
[89] M. Vali, V. Petrovič, L. Y. Pao, and M. Kühn, "Model predictive active power control for optimal structural load equalization in waked wind farms," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 1, pp. 30–44, Jan. 2022, doi: 10.1109/TCST.2021.3053776.
[90] L. Huang et al., "Grid-synchronization stability analysis and loop shaping for PLL-based power converters with different reactive power control," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 501–516, Jan. 2020, doi: 10.1109/TSG.2019.2924295.
[91] H. Liu et al., "Subsynchronous interaction between direct-drive PMSG based wind farms and weak AC networks," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4708–4720, Nov. 2017, doi: 10.1109/TPWRS.2017.2682197.
[92] W. Dong, H. Xin, D. Wu, and L. Huang, "Small signal stability analysis of multi-infeed power electronic systems based on grid strength assessment," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1393–1403, Mar. 2019, doi: 10.1109/TPWRS.2018.2875305.
[93] S. Golestan and J. M. Guerrero, "Conventional synchronous reference frame phase-locked loop is an adaptive complex filter," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1679–1682, Mar. 2015, doi: 10.1109/TIE.2014.2341594.
[94] X. Wang, M. G. Taul, H. Wu, Y. Liao, F. Blaabjerg, and L. Harnefors, "Grid-synchronization stability of converter-based resources—An overview," *IEEE Open J. Ind. Appl.*, vol. 1, pp. 115–134, Aug. 2020, doi: 10.1109/OJIA.2020.3020392.
[95] L. Grüne and A. Rantzer, "On the infinite horizon performance of receding horizon controllers," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2100–2111, Oct. 2008, doi: 10.1109/TAC.2008.927799.
[96] L. Huang et al., "A virtual synchronous control for voltage-source converters utilizing dynamics of DC-link capacitor to realize self-synchronization," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 5, no. 4, pp. 1565–1577, Dec. 2017, doi: 10.1109/JESTPE.2017.2740424.
[97] C. Arghir, T. Jouini, and F. Dörfler, "Grid-forming control for power converters based on matching of synchronous machines," *Automatica*, vol. 95, pp. 273–282, 2018, doi: 10.1016/j.automatica.2018.05.037.
[98] C. Yang, L. Huang, H. Xin, and P. Ju, "Placing grid-forming converters to enhance small signal stability of PLL-integrated power systems," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3563–3573, Jul. 2021, doi: 10.1109/TPWRS.2020.3042741.
[99] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Synchronization and power sharing for droop-controlled inverters in islanded microgrids," *Automatica*, vol. 49, no. 9, pp. 2603–2611, Sep. 2013, doi: 10.1016/j.automatica.2013.05.018.
[100] B. Johnson, S. Dhople, A. Hamadeh, and P. Krein, "Synchronization of parallel single-phase inverters with virtual oscillator control," *IEEE Trans. Power Electron.*, vol. 29, no. 11, pp. 6124–6138, Nov. 2014, doi: 10.1109/TPEL.2013.2296292.
[101] M. Colombino, D. Groß, J.-S. Brouillon, and F. Dörfler, "Global phase and magnitude synchronization of coupled oscillators with application to the control of grid-forming power inverters," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4496–4511, Nov. 2019, doi: 10.1109/TAC.2019.2898549.
[102] Y. Gu and T. C. Green, "Power system stability with a high penetration of inverter-based resources," *Proc. IEEE*, early access, 2022, doi: 10.1109/JPROC.2022.3179826.
[103] H. Gao et al., "Common-mode frequency in converter-integrated power systems: Definition, analysis, and quantitative evaluation," *IEEE Trans. Power Syst.*, vol. 37, no. 6, pp. 4846–4860, Nov. 2022, doi: 10.1109/TPWRS.2022.3150912.
[104] P.-K. Keung, P. Li, H. Banakar, and B. Ooi, "Kinetic energy of wind-turbine generators for system frequency support," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 279–287, Feb. 2009, doi: 10.1109/TPWRS.2008.2004827.
[105] L. Huang, H. Xin, L. Zhang, Z. Wang, K. Wu, and H. Wang, "Synchronization and frequency regulation of DFIG-based wind turbine generators with synchronized control," *IEEE Trans. Energy Convers.*, vol. 32, no. 3, pp. 1251–1262, Sep. 2017, doi: 10.1109/TEC.2017.2675480.
[106] B. Marinescu, O. Gomis-Bellmunt, F. Dörfler, H. Schulte, and L. Sigrist, "Dynamic virtual power plant: A new concept for grid integration of renewable energy sources," *IEEE Access*, vol. 10, pp. 104,980–104,995, Sep. 2022, doi: 10.1109/ACCESS.2022.3205731.
[107] J. C. Willems, "From time series to linear system—Part II. Exact modelling," *Automatica*, vol. 22, no. 6, pp. 675–694, Nov. 1986, doi: 10.1016/0005-1098(86)90005-1.
[108] J. C. Willems, "From time series to linear system—Part III. Approximate modelling," *Automatica*, vol. 23, no. 1, pp. 87–115, Jan. 1987, doi: 10.1016/0005-1098(87)90120-8.