

DATA-DRIVEN CONTROL BASED ON BEHAVIORAL APPROACH: FROM THEORY TO APPLICATIONS IN POWER SYSTEMS

Ivan Markovsky, Linbin Huang, and Florian Dörfler

I. Markovsky is with ICREA, Pg. Lluís Companys 23, Barcelona, and CIMNE, Gran Capitán, Barcelona, Spain (e-mail: imarkovsky@cimne.upc.edu),

L. Huang and F. Dörfler are with the Automatic Control Laboratory, ETH Zürich, 8092 Zürich, Switzerland (e-mails: linhuang@ethz.ch, dorfler@ethz.ch).

Summary

Behavioral systems theory decouples the behavior of a system from its 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-variant 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.

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*: predict the variables in a new experiment,
- » *smoothing*: remove measurement noise from observations and infer hidden/latent variables, and
- » *control*: modify the behavior of some variables by manipulating other variables.

In order 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 (*grey-box*

modeling). Modeling using observed data, possibly incorporating some prior knowledge from the physical laws (that is, black-box and grey-box modeling) is called *system identification*.

System identification is generally applicable and mostly automated (user input may be needed for tuning hyper-parameters). Modeling from first principles in contrast is domain specific and laborious. Identification methods allow also for an accuracy–complexity trade-off, so that simplified approximate models can be obtained, while modeling from first principles delivers exact models. Thus, system identification is often used for modeling complex phenomena, for which models from first principles are difficult or even impossible to obtain. The approximation aspect of system identification, 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 [1]–[3], dual control [4]–[7], and control-regularized identification [8], [9].

Most design methods are *model-based*—they assume a given model. Recently, an alternative paradigm, called *data-driven*, emerged. 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 [10], we adopt the terminology

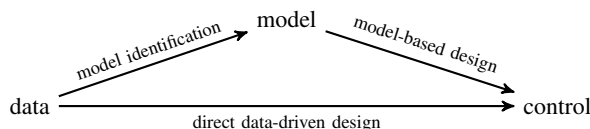


FIGURE 1 The direct data-driven design paradigm aims to achieve a map from data to result (simulated, smoothed, or control signal) without identification of a model of the data-generating process.

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 non-parametric models, such as the data-driven representation presented in the paper, are direct (see Sidebar “Model-free vs model-based methods”).

The shift from indirect to direct design is motivated by

- » *technological advances*: more data, storage, and computational power are available;
- » *system’s complexity*: complex physics, environments, or sensing modalities for which first principle models are not available or are not useful for design; and
- » *new methods*: advances in statistical and computational methods for “big data”.

From a theoretical point of view, however, the main reason for the shift from indirect to direct design is that the two-step procedure of the indirect design may be suboptimal. Data-

minimize cost function(\hat{w})
 subject to \hat{w} is trajectory of model \mathcal{B}
 where \mathcal{B} is identified from given data w_d

FIGURE 2 Indirect (model-based) data-driven design is a bi-level 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 \mathcal{B} . On the inner level, the model \mathcal{B} is identified from given data w_d . The inner problem is also a constrained optimization problem. Its cost function quantifies the fit of w_d by \mathcal{B} , and the constraint restricts \mathcal{B} to a given class of models, *e.g.*, 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.

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—simulated, predicted, or control signal. The indirect model-based approach splits the overall problem into two sequential sub-problems: 1) model identification and 2) model-based design; see Figure 2. The model identification sub-problem 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 sub-problem, 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 special cases, see [3, Section 4], there is no separation principle for system identification and model-based design. Therefore,

an end-to-end direct method may outperform indirect methods.

Additional arguments in favor of the direct approach are incompatibility of system identification and model-based design methods and the need of model structure selection in system identification. 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, grey-box modeling) or using again the data. A typical example of structure selection is choosing the model order [11]. 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 paper are based on the behavioral approach to systems theory [12]. 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 does not have to be labeled into inputs and outputs. Sidebar “Classical vs Behavioral Approach” explains the rationale for the definition of the system as a set, and Section “Behavioral Systems Theory” introduces the notation used in the paper. For an in depth study of the behavioral approach, we recommend the textbook [13], the tutorial paper [14], its follow-up [15], and the survey [16] of the behavioral approach from the data-driven perspective.

Section “Data-Driven Trajectories Interpolation, Extrapolation, and Approximation” 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 which amounts to a non-convex optimization problem, and we approach it by either a sequential (indirect) approach or by convex relaxations giving rise to an ℓ_1 -norm regularization.

This problem setup is further refined to the setting of finite horizon open-loop control in Section “Direct Data-Driven Control”. 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

CLASSICAL VS BEHAVIORAL APPROACH

Traditionally 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 S1.

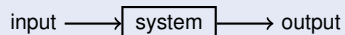


FIGURE S1 The classical view of a dynamical system is a map from an input signal to an output signal.

Many man-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. The input-output map, however, is deficient for modeling many man-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. In order 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 S2), which led to major results, such as the Kalman filter, linear-quadratic Gaussian (LQG) 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.

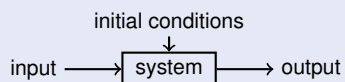


FIGURE S2 Revision of the input/output map view of the system that takes into account the effect of the initial conditions: the output is a function of the input *and* the initial conditions.

Most physical phenomena (for example a mass driven by an external force), however, do not operate as signal 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 then a part of the modeling problem. In order 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 partitioning also leads to issues when interconnecting systems [14]. 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 variables of one system to 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* [12]. 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, 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 [14]

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.

to nonlinear systems, and practical implementation details are discussed, such as hyper-parameter tuning, terminal ingredients for closed-loop stability, or efficient real-time computation.

This direct data-driven control method was successfully used across different applications. Section “Application to Control of Power Electronics Dominated Power Systems” 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 hyper-parameter tuning, and showcase advantages of the direct data-driven approach. The code of the examples shown in the tutorial is available from:

<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 paper 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 paper, 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

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

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

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

A dynamical system postulates which signals w from the universe of signals $(\mathbb{R}^q)^{\mathbb{N}}$ are possible to observe. The signals that are possible to observe are called *trajectories* of the system. The set of all trajectories, denoted by \mathcal{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

$$\text{"a signal } w \text{ is a trajectory of a system } \mathcal{B}." \quad (2)$$

Using the set-theoretic notation of the behavioral approach, this is done by $w \in \mathcal{B}$. What the system \mathcal{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 Sections “Parametric Representations” and “Representation of Restricted Behavior $\mathcal{B}|_T$ ”)

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

$$w, v \in \mathcal{B} \text{ and } \alpha, \beta \in \mathbb{R} \implies \alpha w + \beta v \in \mathcal{B}.$$

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 \mathcal{B} is *time-invariant* if it is invariant under the action of the shift operator

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

that is, $\sigma \mathcal{B} = \mathcal{B}$. 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 \mathcal{L}^q .

Parametric Representations

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

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

Let Π_u be the projection of w on the u variable, that is, $\Pi_u w := u$. Acting on a set, Π_u projects all elements in the set, which results in a new set. The partitioning (3) is an *input/output partitioning* of \mathcal{B} if $\Pi_u \mathcal{B} = (\mathbb{R}^m)^{\mathbb{N}}$, that is, u is a free variable (which means that \mathcal{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).

An *input/state/output representation* of a finite-dimensional LTI system \mathcal{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 :

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

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

$$\mathcal{B}_{ss}(A, B, C, D, \Pi) := \{w = \Pi \begin{bmatrix} u \\ y \end{bmatrix} \in (\mathbb{R}^q)^{\mathbb{N}} \mid \text{there is } x \in (\mathbb{R}^n)^{\mathbb{N}}, \text{ such that (4) holds}\}. \quad (5)$$

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

The parameters of $\mathcal{B}_{ss}(A, B, C, D, \Pi)$ 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 equation (4). For an autonomous system (5) becomes

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

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. An LTI system that is not finite-dimensional does not admit a state-space representation.

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 n of a minimal input/state/output representations of the system \mathcal{B} is invariant of the representation and is called the *order of the system* \mathcal{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 the Sidebar “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 ℓ of the representation*, because it equals the number of lags (or delays) in the equation. The minimal lag over all kernel representations of the system is invariant of the representation and is called the *lag of the system*. The lag ℓ is equal to the *observability index* of the state-space representation (5), *i.e.*, the smallest integer i , for which the extended observability matrix with i block rows

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

becomes full column rank. In general, $\ell \leq n$. In the single-output case, $\ell = n$.

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 $c := (m, \ell, n)$ — number of inputs, lag, and order — and denote with \mathcal{L}_c^q the set of systems with complexity bounded by c , that is, at most m inputs, lag at most ℓ , 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 \mathcal{B} restricted to a time interval.

Representation of Restricted Behavior $\mathcal{B}|_T$

Consider the restriction

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

of an LTI system $\mathcal{B} \in \mathcal{L}^q$ to the interval $[1, T]$. By the linearity of \mathcal{B} , $\mathcal{B}|_T$ 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 $\mathcal{B}|_T$ of a finite-dimensional LTI system in terms of the parameters A, B, C, D of a state-space representation. Then, we show alternative non-parametric representations of $\mathcal{B}|_T$ in terms of a trajectory w_d of the system \mathcal{B} that are used in direct data-driven methods [16].

Using State-Space Representation

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

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

This system of equations can be written more compactly as

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

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

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

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

$$H(0) = D, \quad H(t) = CA^{t-1}B, \quad \text{for } t = 1, 2, \dots \quad (10)$$

It follows that

$$\mathcal{B}|_T = \text{image } M_T(A, B, C, D, \Pi). \quad (11)$$

The state-space representation of the restricted behavior (11) is a map from the initial condition x_{ini} and the input u to a trajectory $w \in \mathcal{B}|_T$. Vice versa, to every $w \in \mathcal{B}|_T$, $T \geq \ell$, correspond unique x_{ini} and u . The representation (11) shows that, for $T \geq \ell$

$$\dim \mathcal{B}|_T = \text{rank } M_T(A, B, C, D, \Pi) = mT + n. \quad (12)$$

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

Formula (12) 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 \mathcal{B} allows, the more complex it is. For LTI systems, the complexity is then determined by the dimension of \mathcal{B} . Formula (12) shows that the dimension of $\mathcal{B}|_T$ is an affine function of T with offset n and slope m . For

KERNEL REPRESENTATION AND INPUT/OUTPUT PARTITIONS

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

$$\mathcal{B} = \ker R(\sigma) := \{w \in (\mathbb{R}^q)^{\mathbb{N}} \mid R(\sigma)w = 0\}, \quad (S7)$$

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

$$\begin{aligned} R(z) &= R_0 + R_1 z + \dots + R_\ell z^\ell \\ &= \begin{bmatrix} R^1(z) \\ \vdots \\ R^k(z) \end{bmatrix} = \begin{bmatrix} R_0^1 + R_1^1 z + \dots + R_{\ell_1}^1 z^{\ell_1} \\ \vdots \\ R_0^k + R_1^k z + \dots + R_{\ell_k}^k z^{\ell_k} \end{bmatrix}. \end{aligned} \quad (S8)$$

large T , meaning $T \geq \ell$, the complexity is therefore dominated by the number of inputs m . For systems with 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 $w_d \in (\mathbb{R}^q)^{T_d}$ of the system \mathcal{B} as given data. (The subscript “d” stands for “data”.) The problem presented is to find conditions under which the data w_d represents the system \mathcal{B} exactly. Later on, we consider the questions of how to design an experiment for collecting data that satisfies the required conditions and methods for dealing with inexact data.

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

$$\mathcal{H}_L(w_d) := \begin{bmatrix} w_d(1) & w_d(2) & \dots & w_d(T_d - L + 1) \\ w_d(2) & w_d(3) & \dots & w_d(T_d - L + 2) \\ \vdots & \vdots & \ddots & \vdots \\ w_d(L) & w_d(L+1) & \dots & w_d(T_d) \end{bmatrix}.$$

The columns of $\mathcal{H}_L(w_d)$, viewed as L -samples long signals, are trajectories of $\mathcal{B}|_L$. Combining this fact with (12), it follows that the rank of $\mathcal{H}_L(w_d)$ is bounded by $\dim \mathcal{B}|_L = mL + n$.

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

$$\text{rank} \begin{bmatrix} \mathcal{H}_L(w_d) & \mathcal{H}_L(w) \end{bmatrix} \leq mL + n. \quad (13)$$

Inequality (13) expresses the system-theoretic fact that the signal w is a trajectory of the system \mathcal{B} that generated w_d , as an algebraic condition—rank deficiency of a matrix constructed from w and w_d . Note that the data-generating system \mathcal{B} is implicit in the Hankel structured and the rank constraint of (13). Therefore, (13) can be used in data-driven signal processing and control for checking the constraint $w \in \mathcal{B}$, where $\mathcal{B} \in \mathcal{L}^q$ is the unknown data-generating system implicitly specified by w_d , without involving a representation of the system \mathcal{B} .

The representation (S7) is called *minimal* if the number of equations k is as small as possible over all kernel representations of \mathcal{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 \mathcal{B} .

Let $\mathcal{B} = \ker R(\sigma)$ be a minimal kernel representation of \mathcal{B} . The partitioning (3) is an *input/output partitioning* of \mathcal{B} if and only if $\begin{bmatrix} Q & -P \end{bmatrix} := R\Pi^{-1}$, with $P \in \mathbb{R}^{p \times p}$ non-singular [S1]. The resulting *input/output representation* is

$$\mathcal{B}_{\text{io}}(P, Q, \Pi) = \{\Pi \begin{bmatrix} u \\ y \end{bmatrix} \mid Q(\sigma)u = P(\sigma)y\}. \quad (S9)$$

REFERENCES

[S1] J. C. Willems, “Paradigms and puzzles in the theory of dynamical systems,” *IEEE Trans. Automat. Contr.*, vol. 36, pp. 259–294, 1991.

For any $w_d \in \mathcal{B}|_{T_d}$ and $L \in [1, T_d]$, we have that

$$\text{image } \mathcal{H}_L(w_d) \subseteq \mathcal{B}|_L. \quad (14)$$

Under the condition, called *generalized persistency of excitation*,

$$\text{rank } \mathcal{H}_L(w_d) = mL + n, \quad (15)$$

for $L > \ell$, due to (12), equality holds in (14) [16], that is,

$$\mathcal{B}|_L = \text{image } \mathcal{H}_L(w_d). \quad (16)$$

We refer to the representation (16) of the restricted behavior $\mathcal{B}|_L$ as a finite-horizon *data-driven representation* of the system \mathcal{B} because it is expressed directly in terms of the raw data w_d .

Under the generalized persistency of excitation condition (15), the rank condition (13) holds with equality. Note that in the special case $L = T$, the matrix in (13) becomes $\begin{bmatrix} \mathcal{H}_T(w_d) & w \end{bmatrix}$, which implies that

$$\mathcal{H}_T(w_d)g = w, \quad (17)$$

for some $g \in \mathbb{R}^{T_d - T + 1}$. Since w is an arbitrary element of $\mathcal{B}|_T$, (17) is equivalent to (16). The vector g parametrizes $\mathcal{B}|_T$ in a similar way as the initial condition x_{ini} and the input u parameterize $\mathcal{B}|_T$ in (11). However, contrary to the parameterization (11), the data-driven parameterization (16) is in general redundant—given $w \in \mathcal{B}|_L$, g is not unique.

Choosing the hyper-parameter L , $\ell + 1 \leq L \leq T$ is exploited in Section “Data-Driven Trajectories Interpolation, Extrapolation, and Approximation” for development of different methods. The most basic method for solving data-driven problems follows from (17), that is, we choose $L = T$. Then, the constraint $w \in \mathcal{B}|_T$ is equivalent to the existence of a solution $g \in \mathbb{R}^{T_d - T + 1}$ of the system of linear equations (17).

The data-driven representation (16) and the generalized persistency of excitation condition (15) can be used in case of data consisting of multiple trajectories w_d^1, \dots, w_d^N by replacing the Hankel matrix $\mathcal{H}_L(w_d)$ with the *mosaic-Hankel matrix* [17]

$$\mathcal{H}_L(w_d^1, \dots, w_d^N) := \begin{bmatrix} \mathcal{H}_L(w_d^1) & \dots & \mathcal{H}_L(w_d^N) \end{bmatrix}.$$

Also, (13) has the following generalization for the case when the data consists of multiple trajectories:

$$\text{rank } \mathcal{H}_L(w_d^1, \dots, w_d^N, w) \leq mL + n.$$

The mosaic-Hankel matrix structure is visualized in Figure 5. Other data structures used in data-driven control (presented later) are the page matrix [18] and the trajectory matrix [19]. They are special cases of the mosaic-Hankel matrix.



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

An alternative result to (15) that guarantees the data-driven representation (16) is presented in [20]. It is historically the first result that gives necessary conditions for a data-driven representation and became known as the *fundamental lemma* in recognition of its importance for system identification and data-driven control [16]. The fundamental lemma was motivated by and has its origin in subspace system identification [21]. 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 co-workers to the fundamental lemma. Although the result was envisaged as input design for identifiability, it opened the path to direct data-driven simulation and control [22] as well as to system-theoretic interpretation of subspace identification methods [23]. An overview of the fundamental lemma and its generalizations is given in Sidebar “The fundamental lemma”.

Transformations Among Representations

Methods for checking 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 is 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 6 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 is 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 parametrizable 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 are analysis problems. In general, going from a less structured to a more structured representation makes the problem harder. An important problem is the transition from the impulse response to an input/state/output representation of the system. This problem was solved in the 1960’s 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 an input/output and input/state/output representations involves choosing an input/output partitioning, see the Sidebar “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, (16) led to new data-driven signal processing and control methods.

We’ve presented three characterizations of the constraint that a finite signal $w \in (\mathbb{R}^q)^T$ is a trajectory of an LTI system \mathcal{B} . First, using a state-space representation $\mathcal{B}(A, B, C, D, \Pi)$ of \mathcal{B}

$$w \in \mathcal{B}|_T \iff \text{there is } x_{\text{ini}} \in \mathbb{R}^n, \text{ such that} \\ w = M_T(A, B, C, D, \Pi) \begin{bmatrix} x_{\text{ini}} \\ u \end{bmatrix}. \quad (19)$$

Second, using a trajectory $w_d \in \mathcal{B}|_{T_d}$ that satisfies the generalized persistency of excitation condition (15) for an $L \in [\ell + 1, T]$,

$$w \in \mathcal{B}|_T \iff \text{rank} \begin{bmatrix} \mathcal{H}_L(w_d) & \mathcal{H}_L(w) \end{bmatrix} \\ = \text{rank } \mathcal{H}_L(w_d). \quad (20)$$

Third, using a trajectory $w_d \in \mathcal{B}|_{T_d}$ that satisfies the generalized persistency of excitation condition (15) for $L = T$,

$$w \in \mathcal{B}|_T \iff \text{there is } g \in \mathbb{R}^{T_d - T + 1}, \text{ such that} \\ w = \mathcal{H}_T(w_d)g. \quad (21)$$

Methods using the state-space representation (19) require system identification, which is typically a nonconvex optimization problem. The direct data-driven methods presented in the following sections use the rank constraint (20) and the system of equations (21). The approach using the rank constraint (20) leads

THE FUNDAMENTAL LEMMA

The problem studied in [20] is to find for a given finite trajectory $w_d \in (\mathbb{R}^q)^T$ of an LTI system \mathcal{B} and a given natural number L , $1 \leq L \leq T_d$, conditions under which the "windows" (trajectory snippets) of length L

$$\begin{bmatrix} w_d(1) \\ \vdots \\ w_d(L) \end{bmatrix}, \begin{bmatrix} w_d(2) \\ \vdots \\ w_d(L+1) \end{bmatrix}, \dots, \begin{bmatrix} w_d(T_d-L+1) \\ \vdots \\ w_d(T_d) \end{bmatrix}, \quad (\text{S18})$$

constructed from the trajectory w_d span the space $\mathcal{B}|_L$ of all possible trajectories of length L , which the system can produce. A compact way of writing that (S18) spans $\mathcal{B}|_L$ is the data-driven representation (16). The problem addressed in [20] is then: "Under what conditions does (16) hold true?"

The solution given in [20] assumes a given input/output partitioning $w = \begin{bmatrix} u \\ y \end{bmatrix}$ of the variables and provides sufficient conditions only: (16) holds true assuming that

- A1: \mathcal{B} is controllable (in the behavioral sense, see [S1, Definition V.1]) and
- A2: u_d is persistently exciting of order $L+n$, that is, $\mathcal{H}_{L+n}(u_d)$ is full row-rank.

Assumption A1 is not verifiable from the data, and assumption A2 requires prior knowledge of the order n of the data-generating system \mathcal{B} . The need to assume input/output partitioning and controllability as well as the sufficiency but not necessity of A1 and A2 make the result of [20] more restrictive than (15). Obtaining conditions for (16) in terms of the input u_d is motivated from the *input design* perspective [S2]–[S5]. Assumption A2 can be used for choosing the input so that the data w_d is guaranteed to ensure (16) for *any* initial condition.

As stated in [20], the extra persistency of excitation of order n in assumption A2 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 in order to be able to deduce that the observed sequences (S18) 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_d than the width of the windows of w_d which are considered."

The original publication [20] 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 A1 and A2 are. These questions are addressed in [S6], where an alternative constructive proof of the fundamental lemma is given. It shows that assumptions A1 and A2 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.

The fundamental lemma was conceived as a theoretical result in system identification; 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 [24],
- » uncontrollable systems [25], [S8], and
- » other model classes: affine [26], linear parameter-varying [S9], linear time-varying [27], flat [S11], delay [S12], finite impulse response Volterra [S13], Wiener-Hammerstein [S14], and bilinear [S15].

REFERENCES

- [S1] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 259–294, 1991.
- [S2] H. van Waarde, "Beyond persistent excitation: Online experiment design for data-driven modeling and control," *IEEE Contr. Systems Lett.*, vol. 6, pp. 319–324, 2021.
- [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, 2021.
- [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, 2021.
- [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," arXiv:2208.00905, 2022.
- [S6] I. Markovsky, E. Prieto-Araujo, and F. Dörfler, "On the persistency of excitation," *Automatica*, 2023.
- [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 Contr. Systems Lett.*, vol. 4, pp. 602–607, 2020.
- [S8] Y. Yu, S. Talebi, H. van Waarde, U. Topcu, M. Mesbahi, and B. Açikmeşe, "On controllability and persistency of excitation in data-driven control: Extensions of Willems' fundamental lemma," in *60th IEEE Conf. on Decision and Contr.*, 2021, pp. 6485–6490.
- [S9] C. Verhoeck, R. Tóth, S. Haesaert, and A. Koch, "Fundamental lemma for data-driven analysis of linear parameter-varying systems," in *60th IEEE Conf. on Decision and Contr.*, 2021, pp. 5040–5046.
- [S10] B. Nortmann and T. Mylvaganam, "Direct data-driven control of linear time-varying systems," arXiv:2111.02342, 2021.
- [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 *60th IEEE Conf. on Decision and Contr.*, 2021, pp. 1484–1489.
- [S12] J. G. Rueda-Escobedo, E. Fridman, and J. Schiffer, "Data-driven control for linear discrete-time delay systems," *IEEE Trans. Automat. Contr.*, vol. 67, no. 7, pp. 3321–3336, 2021.
- [S13] J. Rueda-Escobedo and J. Schiffer, "Data-driven internal model control of second-order discrete Volterra systems," in *59th IEEE Conf. on Decision and Contr.*, 2020, pp. 4572–4579.
- [S14] J. Berberich and F. Allgöwer, "A trajectory-based framework for data-driven system analysis and control," in *European Contr. Conf.*, 2020, pp. 1365–1370.
- [S15] A. Bisoffi, C. De Persis, and P. Tesi, "Data-based stabilization of unknown bilinear systems with guaranteed basin of attraction," *Systems & Contr. Lett.*, vol. 145, p. 104788, 2020.

MODEL-FREE VS MODEL-BASED METHODS

The term “*model-free*” is sometimes used in the literature as a synonym for what we call in this paper “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 (16) 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 (16) uses $\text{image } \mathcal{H}_L(w_d)$ as a model of the data-generating system over the horizon $[1, L]$.

The key question in the model-free vs model-based dilemma is rather how the methods impose constraints on the trajectory \hat{w} (see Figure 2). The parametric representations (5) and (S7) 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 (16), in contrast, “lets the data speak” by not restricting a priori the model’s complexity. The only constraints that (16) imposes on \hat{w} are the ones already present in the data w_d . As shown in Sections “Models for Inexact Data w_d ” and “Direct Data-Driven Control”, this freedom becomes an issue when the data is not exact because, generically, $\text{image } \mathcal{H}_L(w_d) = \mathbb{R}^{qL}$, and therefore (16) does not impose constraints. In this case, constraints should be enforced in order to avoid the trivial model $\hat{\mathcal{B}}|_L = (\mathbb{R}^q)^L$.

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.

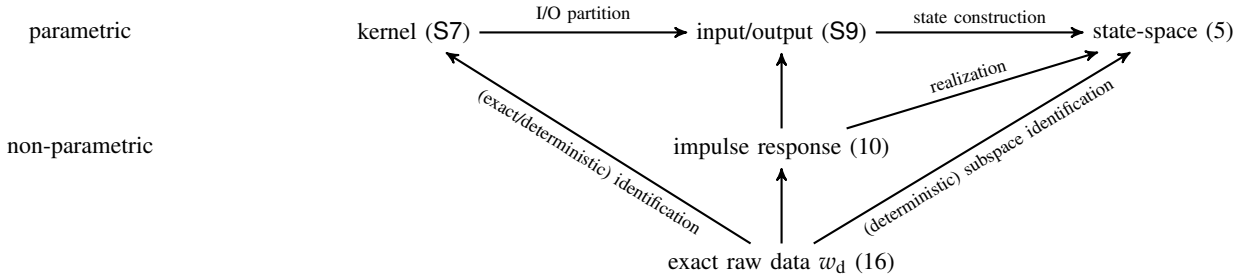


FIGURE 6 Representations of dynamical systems and some transformations among them: The structure of the system that is implied by the representations grows from bottom to top (raw data, nonparametric, parametric) and from left to right (no separation of the variables, input/output partition, state-space). Least structured is 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.

to structured low-rank matrix approximation and completion problems [29], which are nonconvex optimization problems. The approach using the system of equations (21) leads to regularized optimization problems, which are convex [10]. This suggests choosing (21). For $L < T$, however, the rank constraint (20) loosens the generalized persistency of excitation condition. In the extreme $L = \ell + 1$, (20) reduces to parametric (kernel representation) identification of the data-generating system. Thus, the hyper-parameter L in (21) achieves a transition from a model-based representation to the data-driven representation (21).

DATA-DRIVEN TRAJECTORIES INTERPOLATION, EXTRAPOLATION, AND APPROXIMATION

This section presents a generic problem that includes simulation, smoothing, and control as special cases. First, we motivate and formally define the generic problem. Then, we present a direct data-driven solution method based on the data-driven

representation (16). The key assumption is that the data trajectory w_d is exact and satisfies the generalized persistency of excitation condition (15). Finally, we present methods for inexact data w_d , where the inexactness is due to measurement errors, disturbances, or the true data-generating system not being LTI.

Problem Formulation

Interpolation and approximation of functions are basic problems in science and engineering. The solution is sought within a given class of functions. The reader may be familiar with the polynomial, piece-wise linear, and spline interpolation/approximation. The class of functions that we consider here is trajectories of finite-dimensional LTI systems. For example, in 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 interpola-

tion problem where “past” samples of a trajectory are given and “future” samples are missing/to-be-found. In the interpolation problem the given data is 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.

In order 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, \dots, I_K . Similarly, $\mathcal{H}_T(w_d)|_I$ is the submatrix of $\mathcal{H}_T(w_d)$ with row indices I_1, \dots, I_K . In what follows, I_{given} denotes the indices of the *given elements*, and I_{missing} denotes the indices of the *missing elements*.

The interpolation/approximation problem considered

$$\begin{aligned} & \text{minimize over } \hat{w} \quad \|w|_{I_{\text{given}}} - \hat{w}|_{I_{\text{given}}}\|_W^2 \\ & \text{subject to} \quad \hat{w} \in \mathcal{B}|_T \end{aligned} \quad (22)$$

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

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

defined by a positive definite weight matrix W . When W is diagonal the problem is *element-wise weighted*. Larger values of weights in an element-wise weighted problem enforce better approximation of the corresponding data elements. In the limit of weights going to infinity, equality constraints $w|_{I_{\text{exact}}} = \hat{w}|_{I_{\text{exact}}}$ are enforced for the set of elements I_{exact} corresponding to the infinite weights. Thus, equality constraints (or equivalently interpolation points) are specified via the cost function.

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 possibility to recover missing data, approximate inexact data, and interpolate exact data makes problem (22) general. Sidebar “Simulation, smoothing, and control as trajectory interpolation/approximation” shows how classical problems fit into the problem formulation (22).

The constraint $\hat{w} \in \mathcal{B}|_T$ in (22) imposes that \hat{w} is a valid trajectory. In the model-based version of the problem, the system \mathcal{B} is a priori given, while in the data-driven version of the problem \mathcal{B} is implicitly specified by another given trajectory $w_d \in \mathcal{B}|_{T_d}$. We consider two scenarios for the data trajectory w_d :

- 1) w_d is exact, that is, $w_d \in \mathcal{B}|_{T_d}$, where $\mathcal{B} \in \mathcal{L}_c^q$, and
- 2) w_d is inexact, that is, w_d is not a trajectory of a bounded complexity LTI system.

With exact data w_d satisfying the generalized persistency of condition (15), the model-based and the data-driven versions of the interpolant/approximant problem (22) are equivalent. Moreover, the problem is convex, so that 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 meth-

ods does not hold when w_d is inexact. An approximation of the true data-generating system \mathcal{B} by a model $\hat{\mathcal{B}} \in \mathcal{L}_c^q$ 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 Case of Exact Data w_d

We start with the data-driven version of the general interpolation/approximation problem (22) assuming exact data $w_d \in \mathcal{B}|_{T_d}$. The given data is $w|_{I_{\text{given}}}$, i.e., 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 hyper-parameter L , which is an integer in the interval $\ell + 1 \leq L \leq T$. Assuming the generalized persistency of excitation condition (15) and using (13) for the constraint of (22), we obtain

$$\begin{aligned} & \text{minimize over } \hat{w} \quad \|w|_{I_{\text{given}}} - \hat{w}|_{I_{\text{given}}}\|_W^2 \\ & \text{subject to} \quad \text{rank} \begin{bmatrix} \mathcal{H}_L(w_d) & \mathcal{H}_L(\hat{w}) \end{bmatrix} = mL + n, \end{aligned} \quad (27)$$

which is a mosaic-Hankel structured low-rank matrix completion and approximation problem [29]. Due to the rank constraint, (27) is nonconvex. For $L = T$, however, using (17) for the constraint of (22), the problem becomes a linear least-squares problem

$$\text{minimize over } g \quad \|w|_{I_{\text{given}}} - \mathcal{H}_T(w_d)|_{I_{\text{given}}} g\|_W^2. \quad (28)$$

In general, a minimizer g of (28) is not unique. However, any g that minimizes (28) gives rise to the same approximation/interpolation $\hat{w} = \mathcal{H}_T(w_d)g$ of the data $w|_{I_{\text{given}}}$. The non-uniqueness of g is effectively used in the case of noisy data by regularization techniques.

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

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

we obtain a closed-form expression for a solution of (22)

$$\hat{w} = \mathcal{H}_T(w_d) (\sqrt{W} \mathcal{H}_T(w_d)|_{I_{\text{given}}})^\dagger \sqrt{W} w|_{I_{\text{given}}}. \quad (29)$$

If the solution (22) is not unique, the set of all possible solutions is affine and can be characterized explicitly, see [30].

The data-driven interpolation/approximation method (29) has no hyper-parameter. Although the model complexity $c = (m, \ell, n)$ is needed for checking the generalized persistency of excitation condition (15), the method does not depend on it. This suggests that the direct data-driven method (29) has an advantage over indirect methods based on parametric model representations for high-complexity systems. This is confirmed empirically in Sidebar “Data-driven simulation of high-order system”.

Example: Data-Driven Free Fall Prediction

We 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

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 [S1]. Here we show how these problems fit in the interpolation / extrapolation problem (22) for particular choices of the trajectory w . Apart from the input/output partitioning $w = \begin{bmatrix} u \\ y \end{bmatrix}$, we split the time axis into "past" w_{ini} —the first T_{ini} samples—and "future" w_f —the remaining T_f samples. The past/future partition allows us to take into account initial condition in a representation free manner, see Figure S1. Thus, in the problems we specify the four elements of w — u_{ini} , y_{ini} , u_f , and y_f —with their type:

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

A summary of the examples is given in Table S1.

TABLE S1 Simulation, smoothing, and tracking control are interpolation/extrapolation problems (22) for particular choices of the trajectory $w = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \end{bmatrix} \wedge \begin{bmatrix} u_f \\ y_f \end{bmatrix}$, with u_{ini} , y_{ini} , u_f , and y_f missing, inexact, or exact.

	u_{ini}	y_{ini}	u_f	y_f
simulation (S24)	exact	exact	exact	missing
smoothing (S25)	missing	missing	inexact	inexact
control (S26)	exact	exact	missing	inexact

Simulation

In order to define simulation in the behavioral setting, we need to specify 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_{ini} to the trajectory w_f , for which the initial condition has to be specified. The idea is visualized in Figure S1 and formalized in Lemma 1.

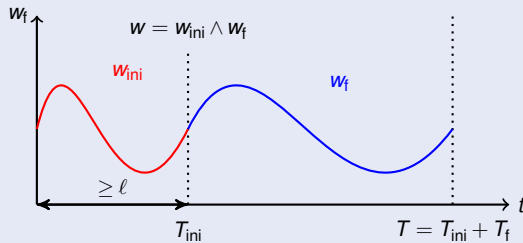


FIGURE S1 The initial condition for a trajectory $w_f \in \mathcal{B}|_{T_f}$ is specified by a pre-fix trajectory $w_{\text{ini}} \in \mathcal{B}|_{T_{\text{ini}}}$ of length $T_{\text{ini}} \geq \ell$.

Lemma 1 (Initial condition specification [22])

Let $\mathcal{B} \in \mathcal{L}^q$ admits an input/output partition $w = \begin{bmatrix} u \\ y \end{bmatrix}$. Then, for any given $w_{\text{ini}} \in \mathcal{B}|_{T_{\text{ini}}}$ with $T_{\text{ini}} \geq \ell$, where ℓ is the lag of \mathcal{B} , and $u_f \in (\mathbb{R}^m)^{T_f}$, there is unique $y_f \in (\mathbb{R}^p)^{T_f}$, such that

$$w_{\text{ini}} \wedge (u_f, y_f) \in \mathcal{B}|_{T_{\text{ini}}+T_f}, \quad (\text{S23})$$

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

The simulation problem is defined as follows: Given a system \mathcal{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 $w_{\text{ini}} \in (\mathbb{R}^q)^{T_{\text{ini}}}$,

$$\text{find } y_f \in (\mathbb{R}^p)^{T_f}, \text{ such that (S23) holds.} \quad (\text{S24})$$

(S24) is an interpolation problem (22) for a partially specified trajectory $w = w_{\text{ini}} \wedge (u_f, y_f)$, with w_{ini} and u_f exact and y_f missing.

Smoothing

The *errors-in-variables Kalman smoothing* problem [S2] is defined as follows: Given an LTI system \mathcal{B} and a "noisy" trajectory $w_f = \bar{w}_f + \bar{w}_f$, where $\bar{w}_{\text{ini}} \wedge \bar{w}_f \in \mathcal{B}|_{T_{\text{ini}}+T_f}$ for some \bar{w}_{ini} and a zero mean, white, Gaussian noise \bar{w}_f with 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:

$$\begin{aligned} &\text{minimize over } \hat{w}_{\text{ini}} \text{ and } \hat{w}_f \quad \|w_f - \hat{w}_f\|_2^2 \\ &\text{subject to } \hat{w}_{\text{ini}} \wedge \hat{w}_f \in \mathcal{B}|_{T_{\text{ini}}+T_f}. \end{aligned} \quad (\text{S25})$$

This problem is an interpolation/approximation problem (22), 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 (22).

Control

In the errors-in-variables Kalman smoothing problem (S25), the data w_f is a "noisy" trajectory of the system \mathcal{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 \mathcal{B}), (S25) 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 unknown initial condition.

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

$$\begin{aligned} &\text{minimize over } \hat{u}_f \text{ and } \hat{y}_f \quad \|y_f - \hat{y}_f\|_2^2 \\ &\text{subject to } w_{\text{ini}} \wedge (\hat{u}_f, \hat{y}_f) \in \mathcal{B}|_{T_{\text{ini}}+T_f}. \end{aligned} \quad (\text{S26})$$

(S26) is an interpolation/approximation problem (22) 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 Section "Direct Data-Driven Control".

REFERENCES

- [S1] I. Markovsky, "A missing data approach to data-driven filtering and control," *IEEE Trans. Automat. Contr.*, vol. 62, pp. 1972–1978, April 2017.
- [S2] I. Markovsky and B. De Moor, "Linear dynamic filtering with noisy input and output," *Automatica*, vol. 41, no. 1, pp. 167–171, 2005.

DATA-DRIVEN SIMULATION OF HIGH-ORDER SYSTEM

This sidebar compares the direct method (29) and an indirect method, that is, system identification 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 \mathcal{B} is a random single-input single-output LTI system of order $n = 100$ (generated in Matlab with `B = drss(100)`), the data w_d is a random trajectory of \mathcal{B} with $T_d = 319$ samples, and $T_f = 10$, that is, we aim to compute the first 10 samples of the step response.

In order to set up (22) for step response simulation, we choose zero initial conditions $w_{\text{ini}} = 0$ with $T_{\text{ini}} = n$ and constant unit input $u_t = 1$. The estimate of the first T_f samples of the step response of \mathcal{B} is obtained then in the block \hat{y}_t of \hat{w} . We verify that the data w_d satisfies the generalized persistency of excitation condition (15) with $L = n + T_f = 110$, so that the data-driven representation (16) holds true. In order to verify that the result computed by (29) is exact, we obtain the first 10 samples \bar{s} of the step response by simulation of \mathcal{B} and compute the error $e = \|\bar{s} - \hat{y}_t\|$. 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 model order or use a larger data set.

Note that a minimal state-space representation $\mathcal{B}_{\text{ss}}(A, B, C, D)$ of \mathcal{B} has $(n+1)^2 = 10201$ parameters. Although this number can be reduced to $2n+1 = 201$ by using a canonical parametrization, subspace identification methods generally require at least $5n$ data samples. Indeed, the maximal model order accepted by `n4sid` is 59. The computed model $\hat{\mathcal{B}}$ of order 59, however, is not exact. Consequently, the step response \hat{s} computed by simulation of $\hat{\mathcal{B}}$ has a nonzero error $e = \|\bar{s} - \hat{s}\|$. The error is highly dependent on the data-generating system \mathcal{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 (16), the minimum number of samples is [16], [31]

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

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_{\min} .

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 in order to collect enough data?” Let N be the number of experiments and let $\mathcal{Y} = \{y_d^1, \dots, y_d^N\}$ be the observed data. It turns out that $N = 5$ trajectories are sufficient, provided that their initial conditions $(y_{\text{ini}}^i, v_{\text{ini}}^i)$, $i = 1, \dots, N$ are linearly independent. The prediction \hat{y} of y is constructed as a linear combination of the observed trajectories

$$\hat{y} := \underbrace{\begin{bmatrix} y_d^1 & \dots & y_d^N \end{bmatrix}}_{Y_d} \begin{bmatrix} g_1 \\ \vdots \\ g_N \end{bmatrix}, \quad (30)$$

where g is obtained from the system of linear equations

$$\begin{bmatrix} y_{\text{ini}}^1 & \dots & y_{\text{ini}}^N \\ v_{\text{ini}}^1 & \dots & v_{\text{ini}}^N \end{bmatrix} g = \begin{bmatrix} y_{\text{ini}} \\ v_{\text{ini}} \end{bmatrix}. \quad (31)$$

matching the initial conditions.

The system of equations (31), however, is underdetermined (4 equations, 5 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

$$\begin{bmatrix} 1 & \dots & 1 \end{bmatrix} g = 1. \quad (32)$$

The free fall phenomenon is an affine time-invariant system $\mathcal{A} \in (\mathbb{R}^2)^N$, that is $\mathcal{A} = y_{\text{fixed}} + \mathcal{B}$ for some $y_{\text{fixed}} \in (\mathbb{R}^2)^N$ and $\mathcal{B} \in \mathcal{L}^2$. As shown in [26], the constraint (32) is a way of generalizing the data-driven representation (16) to affine systems.

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

The compared methods are 1) direct prediction, that is, solve (31) subject to (32) and substitute the solution g in (30) 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 $\hat{\mathcal{B}}$ from the data \mathcal{Y} , estimate the initial conditions, and compute the prediction \hat{y} using the model $\hat{\mathcal{B}}$ and the estimated initial conditions.

Figure 8 shows the relative averaged (over 100 Monte-Carlo runs) estimation error $e := 100\% \|y - \hat{y}\|_2 / \|y\|_2$, as a function of the noise level. The data \mathcal{Y} consists of $N = 20$ experiments. Both the error and the noise level are given in 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 marginally stable (has a pole at 1) — a problematic case for standard identification methods.

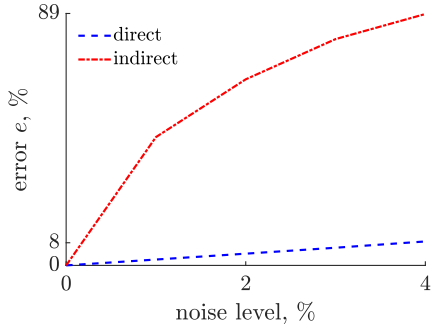


FIGURE 8 Direct and indirect data-driven simulation of free fall in the presence of noise: The relative average estimation error e of the direct method is uniformly better for all noise levels than the one of the indirect method.

Models for Inexact Data w_d

By “ w_d inexact” we mean that there is no bounded complexity LTI system \mathcal{B} for which w_d is a trajectory. Prior knowledge about w_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:

- 1) *errors-in-variables*—additive measurement noise on w_d ,
- 2) *auto-regressive moving-average exogenous (ARMAX)* — unobserved disturbances acting on the system and measurement noise on the output y_d , and
- 3) *nonlinear time-varying system*—the data-generating system is not bounded complexity LTI.

In the errors-in-variables model, the system \mathcal{B} is bounded complexity LTI, however, w_d is observed with measurement noise that is modeled as a stochastic process. In the ARMAX model, the system \mathcal{B} has an additional unobserved input, called *disturbance*, and the output y_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_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 (22) in the *errors-in-variables* setup $w_d = \bar{w}_d + \tilde{w}_d$, where $\bar{w}_d \in \mathcal{B}|_{T_d}$ is the true value and \tilde{w}_d is a zero mean white Gaussian measurement noise [32]. We assume that the model complexity $c = (m, \ell, n)$ is a priori given. Thus, the prior knowledge is concisely written as $\mathcal{B} \in \mathcal{L}_c^q$. The maximum-likelihood estimation problem for the missing data $w|_{I_{\text{missing}}}$ with prior knowledge $\mathcal{B} \in \mathcal{L}_c^q$ is

$$\begin{aligned} & \text{minimize} \quad \text{over } \hat{w}, \hat{w}_d, \hat{\mathcal{B}} \quad \|w|_{I_{\text{given}}} - \hat{w}|_{I_{\text{given}}}\|_W^2 \\ & \quad \quad \quad + \|w_d - \hat{w}_d\|_{W_d}^2 \\ & \text{subject to} \quad \hat{w} \in \hat{\mathcal{B}}|_T, \hat{w}_d \in \hat{\mathcal{B}}|_{T_d}, \text{ and } \hat{\mathcal{B}} \in \mathcal{L}_c^q, \end{aligned} \quad (33)$$

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

Using (13) with $L = \ell + 1$, (33) becomes a mosaic-Hankel structured low-rank approximation and completion problem

$$\begin{aligned} & \text{minimize} \quad \text{over } \hat{w}, \hat{w}_d \quad \|w|_{I_{\text{given}}} - \hat{w}|_{I_{\text{given}}}\|_W^2 + \|w_d - \hat{w}_d\|_{W_d}^2 \\ & \text{subject to} \quad \text{rank} \begin{bmatrix} \mathcal{H}_{\ell+1}(\hat{w}_d) & \mathcal{H}_{\ell+1}(\hat{w}) \end{bmatrix} \leq m(\ell+1) + n. \end{aligned}$$

The problem is nonconvex. Local optimization methods can be used for solving the mosaic-Hankel structured low-rank approximation and completion problem, see [33]–[35], however, these methods involve a kernel representation of $\hat{\mathcal{B}}$.

Another approach for solving (33) is the *two-step procedure*:

- 1) preprocess w_d , aiming to remove the noise, and
- 2) using the “cleaned” signal \hat{w}_d , find $w|_{I_{\text{missing}}}$.

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

$$\begin{aligned} & \text{minimize} \quad \text{over } \hat{w}_d, \hat{\mathcal{B}} \quad \|w_d - \hat{w}_d\|_{W_d}^2 \\ & \text{subject to} \quad \hat{w}_d \in \hat{\mathcal{B}}|_{T_d} \text{ and } \hat{\mathcal{B}} \in \mathcal{L}_c^q. \end{aligned} \quad (34)$$

Problem (34) is a system identification problem. Similarly to (33), (34) can be also 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 (22) with $\mathcal{B} = \hat{\mathcal{B}}$. Thus, the two-step procedure is a model-based approach for solving (33):

- 1) *model identification*: using w_d , identify a model $\hat{\mathcal{B}}$, and
- 2) *model-based interpolation/approximation*: using $\hat{\mathcal{B}}$ and $w|_{I_{\text{given}}}$, estimate $w|_{I_{\text{missing}}}$.

In general, the two-step procedure is suboptimal. When $\dim I_{\text{given}} \leq mT + n$, however, the problem decouples, and the two-step procedure is optimal, that is, the solution of (33) coincides with the solution of (34) followed by (22).

A heuristic for preprocessing w_d is to do rank- $mT + n$ approximation of $\mathcal{H}_T(w_d)$ 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 $\hat{\mathcal{B}}$ of \mathcal{B} .

Algorithm 1 Data-driven interpolation with low-rank approximation preprocessing.

Input: $w_d, I_{\text{given}}, w|_{I_{\text{given}}}, m$, and n .

- 1: Compute the SVD: $\mathcal{H}_T(w_d) = U\Sigma V^\top$.
- 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 $\hat{w} := P(P|_{I_{\text{given}}})^\dagger w|_{I_{\text{given}}}$.

Output: \hat{w} .

Other methods for solving the missing data estimation with

noisy data are nuclear norm relaxation [36]

$$\begin{aligned} \text{minimize over } \hat{w}_d \text{ and } \hat{w} \quad & \|w|_{I_{\text{given}}} - \hat{w}|_{I_{\text{given}}}\|_W^2 \\ & + \|w_d - \hat{w}_d\|_{W_d}^2 + \gamma \left\| \begin{bmatrix} \mathcal{H}_T(\hat{w}_d) & \mathcal{H}_T(\hat{w}) \end{bmatrix} \right\|_* \end{aligned}$$

and ℓ_1 -norm regularization of (22)

$$\text{minimize over } g \quad \|w|_{I_{\text{given}}} - \mathcal{H}_T(w_d)|_{I_{\text{given}}}\|_W^2 + \lambda \|g\|_1. \quad (35)$$

The latter is proposed in [10] as a relaxation for solving a related data-driven control problem (see section “Direct Data-Driven Control”) and is used for data-driven interpolation in [30].

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 data-base for system identification DAISY [37], see [16, Sect. 4.5] and [30, Sect. 5]. In Section “Application to Control of Power Electronics Dominated Power Systems”, we validate the performance of data-driven control methods presented in the following section on applications in the energy domain.

Summary: This section demonstrates the synergy between theory and algorithms. The basic direct data-driven interpolation/approximation is presented for LTI systems. The solution (29) is given in closed form and has no hyper-parameters. In case of noisy data, the maximum-likelihood estimation problem is a mosaic-Hankel structured low-rank approximation, which is non-convex. Convex relaxation based on ℓ_1 -norm regularization is effective, however, it has a hyper-parameter controlling the complexity–accuracy trade-off that needs ad hoc selection.

Applied to the free fall example, the data-driven prediction makes no use of the Newton’s second law, the law of gravity, and the law of friction that are needed for modeling of the free fall from first principles. Also, the identification of the physical parameters is avoided. Prediction of the free fall trajectory without first principles modeling is surprising. Moreover, the 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.

DIRECT DATA-DRIVEN CONTROL

The field of data-driven control has a long and rich history, and it is has been gaining significant traction in the last years due to recent advances in technology (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. To provide a rough classification in the terminology of adaptive control [38], different approaches can be classified as *indirect*, i.e., based on system identification (SysID) followed by model-based design, versus *direct* when by-passing models. Another distinction is *certainty-equivalence* versus *robust* design depending on whether uncertainty is taken into account.

In what follows, we focus on methods surrounding the *Fundamental Lemma* [20] resulting in the data-driven representation

(16) of the restricted behavior parametrizing the set of all finite-length trajectories by a matrix data time series. We do not provide a comprehensive coverage of the literature but refer the reader to the recent survey [39]. In particular, we will not cover explicit control policies derived from the fundamental lemma, c.f., [40]–[42]. Rather we focus on an implicit receding horizon control approach as in classic *model predictive control* (MPC) augmented with *moving horizon estimation* (MHE) [43], [44].

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 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.”
J. C. Willems, [45]

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, i.e., representation-free: the set of trajectories can be spanned by a parametric model or mere data as in (16).

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 towards direct and robust approaches. In particular, we begin with a refined version of the data-driven approximation problem (22). Consider the linear quadratic (LQ) optimal tracking control problem

$$\begin{aligned} \text{minimize}_{u_f, y_f} \quad & \sum_{t=1}^{T_f} \|y_f(t) - y_r(t)\|_Q^2 + \|u_f(t) - u_r(t)\|_R^2 \\ \text{subject to} \quad & (u_{\text{ini}}, y_{\text{ini}}) \wedge (u_f, y_f) \in \mathcal{B}|_{T_{\text{ini}}+T_f} \end{aligned} \quad (36)$$

on a finite horizon $T_f > 0$, where $w_r = (u_r, y_r) \in \mathbb{R}^{qT_f}$ is a user-defined reference trajectory (not necessarily in $\mathcal{B}|_{T_f}$), and $w_f = (u_f, y_f) \in \mathbb{R}^{qT_f}$ is the future trajectory of length $T_f \geq 1$ 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 $\|e\|_Q = \sqrt{e^T Q e}$ is a (semi-)norm for $Q \succeq 0$. Finally, $w_{\text{ini}} = (u_{\text{ini}}, y_{\text{ini}})$ is a given pre-fix trajectory of length $T_{\text{ini}} \geq \ell$ setting the initial condition x_{ini} akin to MHE [22]. Indeed, when resorting to a state-space representation, x_{ini} can be uniquely recovered from

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

since $\mathcal{O}_{T_{\text{ini}}}(A, C)$ has full rank for $T_{\text{ini}} \geq \ell$; see also Lemma 1.

The LQ control problem (36) can be solved by a variety of methods provided that a model of \mathcal{B} is available [46], typically

a state-space model as in (11). In this case, (36) becomes

$$\begin{aligned} & \text{minimize} \quad \text{over } u_f, y_f, x_{\text{ini}} \quad \|y_f - y_r\|_Q^2 + \|u_f - u_r\|_R^2 \\ & \text{subject to} \quad \begin{bmatrix} y_{\text{ini}} \\ y_f \end{bmatrix} = \mathcal{O}_{T_{\text{ini}}+T_f}(A, C)x_{\text{ini}} \\ & \quad \quad \quad + \mathcal{E}_{T_{\text{ini}}+T_f}(A, B, C, D) \begin{bmatrix} u_{\text{ini}} \\ u_f \end{bmatrix}, \end{aligned} \quad (37)$$

where (with slight abuse of notation) we redefined Q and R as $\text{blkdiag}(Q, \dots, Q)$ and $\text{blkdiag}(R, \dots, R)$, respectively. The formulation (37) is standard in MPC and MHE. We now focus on the LQ problem if no model and only raw data is 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 $w_d = (u_d, y_d)$ collected offline and assembled in a Hankel matrix $\mathcal{H}_{T_{\text{ini}}+T_f}(w_d)$. According to $w_{\text{ini}} = (u_{\text{ini}}, y_{\text{ini}})$ and $w_f = (u_f, y_f)$, and permute and partition the Hankel matrix as

$$\begin{bmatrix} w_{\text{ini}} \\ w_f \end{bmatrix} \sim \begin{bmatrix} u_{\text{ini}} \\ u_f \\ y_{\text{ini}} \\ y_f \end{bmatrix}, \quad \mathcal{H}_{T_{\text{ini}}+T_f}(w_d) \sim \begin{bmatrix} U_p \\ U_f \\ Y_p \\ Y_f \end{bmatrix} = \begin{bmatrix} \mathcal{H}_{T_{\text{ini}}+T_f}(u_d) \\ \mathcal{H}_{T_{\text{ini}}+T_f}(y_d) \end{bmatrix},$$

where \sim denotes similarity under a permutation of the rows. As an identifiability condition, we assume that the data $w_d = (u_d, y_d)$ is persistently exciting, i.e., $\mathcal{H}_{T_{\text{ini}}+T_f}(w_d)$ is of rank $m \cdot (T_{\text{ini}} + T_f) + n$, where $T_{\text{ini}} \geq \ell$ and (ℓ, 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 (36) we seek a linear multi-step predictor, i.e., a matrix K relating past as well as future inputs and outputs:

$$y_f = \underbrace{\begin{bmatrix} K_p & K_f \end{bmatrix}}_{=K} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \end{bmatrix}. \quad (38)$$

Alternatively, of course a single-step predictor (e.g., in the form of a state-space model) can be identified, and a multi-step predictor can be built recursively as in (37). We refer to [47] for a general discussion on multi-step versus single-step predictors.

The multi-step predictor K can also be found from data by replacing the variables $(u_{\text{ini}}, y_{\text{ini}}, u_f, y_f)$ in (38) by Hankel matrix data (U_p, Y_p, U_f, Y_f) and solving for K in a least-square sense

$$K = \underset{\hat{K}}{\text{argmin}} \left\| Y_f - \hat{K} \cdot \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \right\|_F = Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^+, \quad (39)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The multi-step predictor (38)–(39) is well known in the context of *subspace predictive control* [48], [49]; see the Sidebar Subspace Predictive Control (SPC). It is educational to compare this purely data-based multi-step predictor to the model-based predictor in (37) parameterized by A, B, C , and D . For exact data, we have that $\text{rank}(K_p) = n$ (assuring LTI behavior of desired complexity) and a lower block-triangular zero pattern of K_f (assuring causality).

Thus, the indirect approach (least-square ID (39) followed by LQ control (36)) can be formalized as a *bi-level* program:

$$\begin{aligned} & \text{minimize} \quad \text{over } u_f, y_f \quad \|y_f - y_r\|_Q^2 + \|u_f - u_r\|_R^2 \\ & \text{subject to} \quad y_f = K \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \end{bmatrix} \end{aligned} \quad (40)$$

$$K = \underset{\hat{K}}{\text{argmin}} \left\| Y_f - \hat{K} \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \right\|_F$$

We now present a direct data-driven approach to LQ control. The fundamental lemma implies that the concatenated initial and future trajectory $w := w_{\text{ini}} \wedge w_f \in \mathcal{B}|_{T_{\text{ini}}+T_f}$ lies in the image of $\mathcal{H}_{T_{\text{ini}}+T_f}(w_d)$. Thus, $w = \mathcal{H}_{T_{\text{ini}}+T_f}(w_d)g$ for some g , and the LQ control problem (36) can be posed as

$$\begin{aligned} & \text{minimize} \quad \text{over } u_f, y_f, g \quad \|y_f - y_r\|_Q^2 + \|u_f - u_r\|_R^2 \\ & \text{subject to} \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ y_f \end{bmatrix}. \end{aligned} \quad (42)$$

This direct LQ control formulation has been proposed in [22]. Its interpretation is intuitive from the perspective of dictionary learning. The columns of the Hankel matrix $\mathcal{H}_{T_{\text{ini}}+T_f}(w_d)$ serve as a trajectory library spanning $\mathcal{B}|_{T_{\text{ini}}+T_f}$. These are then linearly combined through the vector g to match the pre-fix trajectory $w_{\text{ini}} = (u_{\text{ini}}, y_{\text{ini}})$ and to optimally synthesize the future control trajectory $w_f = (u_f, y_f)$; see Figure 9 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, e.g., in the context of motion primitives in robotics for motion planning [50] and predictive control [51].

Of course, all formulations (36)–(42) are identical for exact data. For inexact data, the relations among these formulations have been studied in [39], [53]–[56], and we will resume this discussion when it comes to suitable regularizations.

A comparison of the direct and indirect problem formulations, reveals the usual pros and cons which have often been elaborated upon [3]. For example, the indirect approach consists of modular and well understood steps, but it requires model selection, pre-/post-processing 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 (42) is computationally more involved (more decision variables and less structured matrices), which reveals a strength of parametric models: they are compact, cleaned-up, and de-noised representations. However, the direct approach is end-to-end, arguably simple, and it holds the promise to extend beyond deterministic LTI systems; to be adaptive (i.e., the data matrices could be updated online); and to improve over the sequential bi-level approach (40) (since there is no separation principle for the nested optimization problems). It is the starting for the expanding literature on data-enabled predictive control presented next, and we will see how to robustify it in case of noisy data or nonlinear dynamics.

SUBSPACE PREDICTIVE CONTROL (SPC)

By substituting the explicit least-square solution (39) into (40) one arrives at the SPC problem formulation coined in [48]:

$$\begin{aligned} & \text{minimize over } u_f, y_f \quad \|y_f - y_r\|_Q^2 + \|u_f - u_r\|_R^2 \\ & \text{subject to } y_f = Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger \begin{bmatrix} u_{ini} \\ y_{ini} \\ u_f \end{bmatrix}. \end{aligned} \quad (\text{S41})$$

In case of inexact data, SPC is often augmented with further pre- and post-processing of the data, e.g., 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, it is easy to implement, and has demonstrated excellent performance in numerous case studies; see [49] for a review.

In relation to Data-Enabled Predictive Control (DeePC) (43), the main difference is that SPC is focused on the predictor (S41), whereas DeePC (43) is focused on the prediction itself. Hence, the de-noising and robustification schemes are different (e.g., low-rank approximation versus regularization). Tough, the two methods become aligned when using the projection-based regularizer (50) with sufficiently large regularization coefficient.

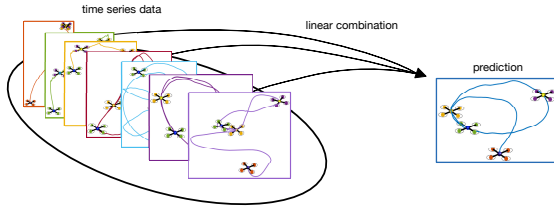


FIGURE 9 Schematic illustration of the direct data-driven LQ control approach (42) to synthesize the motion of a quadcopter. Trajectories from a library are linearly combined to match the pre-fix trajectory and to optimally synthesize the future control trajectory. We refer to [52] for an implementation of this case study.

Data-enabled predictive control

Data-driven approaches to MPC date back to *dynamic matrix control* [57], [58] and the previously reviewed SPC [48], [49]. We refer [59]–[61] for recent surveys and contemporary learning-based MPC approaches. Here, our starting point is the formulation (42) proposed for deterministic LTI systems in [22].

In 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 (42). However, this approach is doomed to fail. This can be intuitively understood from the perspective of dictionary learning alluded to in Figure 9. The optimal control trajectory is synthesized by linearly combining columns of the Hankel matrix $\mathcal{H}_{T_{ini}+T_f}(w_d)$ which spans $\mathcal{B}|_{T_{ini}+T_f}$. However, a superposition of trajectories from $\mathcal{B}|_{T_{ini}+T_f}$ is again a valid trajectory of $\mathcal{B}|_{T_{ini}+T_f}$ only for deterministic LTI systems. What is even more detrimental in a practical setting is that a Hankel matrix $\mathcal{H}_{T_{ini}+T_f}$ built from noisy or nonlinear data will almost surely have full rank, and thus any future trajectory (u_f, y_f) is feasible for (42). As a result, the predicted trajectory can be arbitrarily optimistic and non-realizable when implemented on the real system. Further, the data (u_{ini}, y_{ini}) collected online (before implementing an instance of the optimal control (42)) 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. *Data-Enabled Predictive Control* (DeePC) [62] is a robustified version of direct data-driven LQ control (42):

$$\begin{aligned} & \text{minimize over } u_f, y_f, g, \sigma_{u_{ini}}, \sigma_{y_{ini}} \quad \|y_f - y_r\|_Q^2 + \|u_f - u_r\|_R^2 \\ & \quad + \lambda_{u_{ini}} \|\sigma_{u_{ini}}\|_2^2 + \lambda_{y_{ini}} \|\sigma_{y_{ini}}\|_2^2 + \lambda_g \cdot h(g) \\ & \text{subject to } \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{ini} + \sigma_{u_{ini}} \\ y_{ini} + \sigma_{y_{ini}} \\ u_f \\ y_f \end{bmatrix} \text{ and } (u_f, y_f) \in \mathcal{U} \times \mathcal{Y} \end{aligned} \quad (43)$$

Let us walk slowly through the DeePC problem formulation (43).

First, recall that $w_{ini} = (u_{ini}, y_{ini})$ are online measurements used to estimate an initial condition via the constraint equations $U_p g = u_{ini}$ and $Y_p g = y_{ini}$ in the certainty-equivalence LQ control (42). However, for inexact data these constraints may not be feasible. Hence, DeePC (43) opts for least-square estimation akin to MHE and softens these constraints with the slack variables $\sigma_{u_{ini}}$ and $\sigma_{y_{ini}}$, which are penalized in the cost with nonnegative coefficients $\lambda_{u_{ini}}, \lambda_{y_{ini}}$. Different variations of the penalty are conceivable, e.g., weighting the norms by inverse noise covariances, disregarding $\sigma_{u_{ini}}$ in absence of 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 (42) with input and output constraints $u_f \in \mathcal{U}$ and $y_f \in \mathcal{Y}$, respectively. These can account for, e.g., input saturation, operational limits, or terminal constraints. The latter are needed for closed-loop stability when implementing (43) as a receding horizon predictive controller [43], [44]; see Figure 10. We will return to these points later in Section “DeePC: closed-loop & implementation aspects”.

Third, aside from uncertainty in (u_{ini}, y_{ini}) affecting the constraints *additively*, the certainty-equivalence LQ problem (42) 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 pre-processing the trajectory library (e.g., by seeking a low-rank approximation of $\mathcal{H}_{T_{ini}+T_f}(w_d)$), but in the spirit of direct data-driven control – seeking an online decision based on raw data – DeePC opts for regularizing the problem (42). 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 below.

Last, it is instructive to recall the involved time scales and data sets when implementing DeePC (43) or any of the approaches (36)–(42). We refer to Figure 10 for a schematic illustration. The data (u_d, y_d) of length T_d is collected offline and directly parametrizes the optimal control problem via the matrices U_p, Y_p, U_f , and Y_f . In a model-based setting, the offline data is 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 (43) is repeatedly called upon for estimation and prediction, and the data sets (u_{ini}, y_{ini}) and (u_f, y_f) are updated accordingly. Of course, a natural (albeit non-trivial) extension is to update the data matrices online in the spirit of adaptive control.

The role of regularization in DeePC

Observe that the DeePC formulation (43) can be compactified by eliminating the variables $u_f, y_f, \sigma_{u_{ini}}, \sigma_{y_{ini}}$:

$$\begin{aligned} & \text{minimize over } g \quad \|Y_f g - y_r\|_Q^2 + \|U_f g - u_r\|_R^2 \\ & \quad + \lambda_{y_{ini}} \|Y_p g - y_{ini}\|_2^2 + \lambda_{u_{ini}} \|U_p g - u_{ini}\|_2^2 + \lambda_g \cdot h(g) \\ & \text{subject to} \quad (U_f g, Y_f g) \in \mathcal{U} \times \mathcal{Y} \end{aligned} \quad (44)$$

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, (44) takes the form of a regularized regression problem:

$$\text{minimize over } g \quad \|\mathcal{H}_{T_{ini}+T_f}(w_d)g - w_{r,ini}\|_P^2 + \lambda_g \cdot h(g) \quad (45)$$

The compact formulation (45) provides a regression perspective on DeePC illustrated in Figure 11, and it motivates the application of Bayesian, non-parametric, or robust regression methods. In particular, we can draw from the vast literature on regularized regression to derive, inspire, or interpret different regularizers $h(g)$. The Sidebar Roles of Regularization motivates three often advocated dominant reasons for why to regularize:

- » aid the optimization by strong convexification;
- » condition models on prior knowledge; and
- » robustify the problem formulation.

For the DeePC problem (43) 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 [39] for a more detailed account.

Regularization derived from pre-processing: For inexact data, the matrix $\mathcal{H}_{T_{ini}+T_f}(w_d)$ 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 pre-processed via structured low-rank approximation. Formally, this can be posed as a bi-level optimization problem: solve the

optimal control problem subject to pre-processing of the data:

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

This non-convex bi-level problem can be formally reduced and convexified. We refer to [54] for details and provide an informal account. Since the processed data matrix $\mathcal{H}_{T_{ini}+T_f}(\hat{w}_d^*)$ 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: $\|g\|_0 \leq m \cdot (T_{ini} + T_f) + n$. This constraint can be relaxed to an ℓ_1 -norm and lifted to the objective (with sufficiently large penalty) resulting in

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

for $\lambda_g \geq 0$ sufficiently large. Next, the inner problem is relaxed by dropping the rank constraint so that $\hat{w}_d^* = w_d$, i.e., the data is taken at face value. Hence, we arrive at the regularized problem (45) with $h(g) = \|g\|_1$. In summary, an ℓ_1 -norm regularizer can be interpreted as a surrogate for pre-processing the data via a low-rank approximation to select the model complexity.

Regularization derived from least-square SysID: Recall the indirect LQ control (40). Note that the explicit multi-step predictor (38)–(39) can be written as $y_f = Y_f g$ with g satisfying

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u_f \end{bmatrix} \quad \text{and} \quad 0 = \left\| \left(I - \underbrace{\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}}_{:=\Pi} \right) g \right\|_p. \quad (49)$$

In (49) we identify the first three constraint equations of the data-driven LQ problem (42) and an orthogonality constraint with the projector Π assuring 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

$$h(g) = \|(I - \Pi)g\|_p \quad (50)$$

can be added to problem (42): indeed, for perfect data, all feasible solutions g to (42) 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 bi-level (SysID followed by control) problem (40). Hence, unlike norm-based regularizers $\|g\|_p$ the projection $I - \Pi$ assures consistency: it does not bias the solution obtained for perfect data. We refer to [54] for further details, to [56] for an alternative derivation, and to [63] for an application of this regularizer to the infinite-horizon LQ regulator.

The above two regularizers can be interpreted in Bayesian sense, see the Sidebar Roles of Regularization, i.e., 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 is either pre-processed according to a linear model or such a model is identified via least squares. The magnitude of the regularization coefficient λ_g ensures to

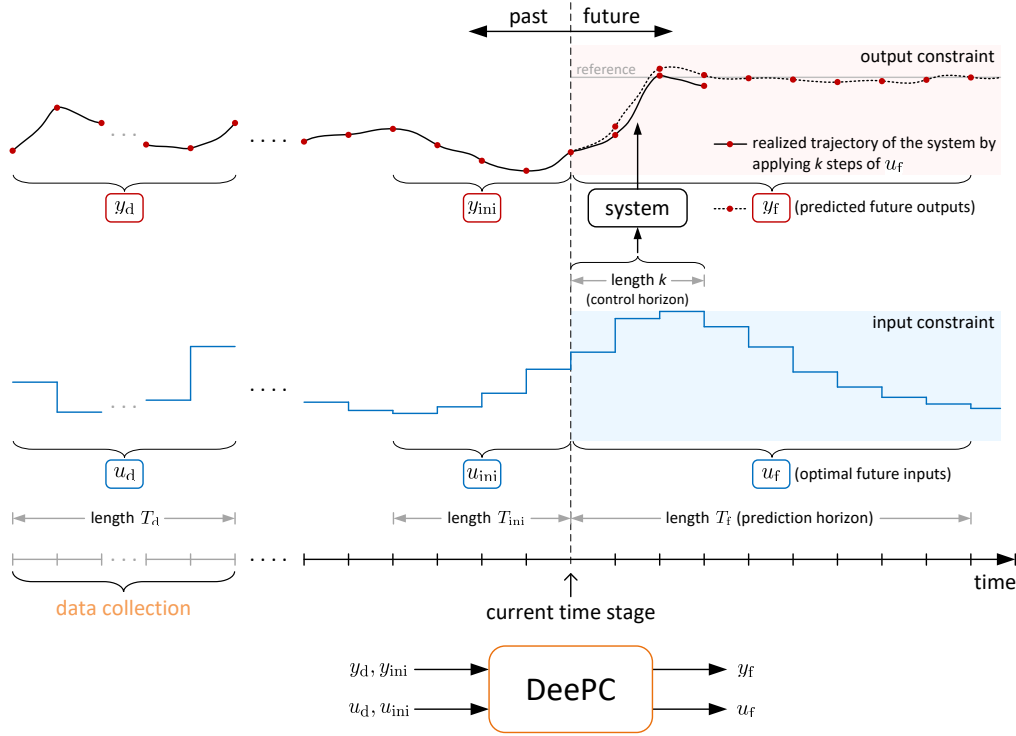


FIGURE 10 Illustration of the different time scales and data sets when implementing DeePC (43). In the offline stage, the data (u_d, y_d) of length T_d is collected to build the matrices U_p , Y_p , U_f , and Y_f . In the online stage, the last T_{ini} measurements (u_{ini}, y_{ini}) are used to initialize 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 (43) is repeatedly called upon after k time steps (k being the control horizon), and the data sets (u_{ini}, y_{ini}) and (u_f, y_f) are updated.

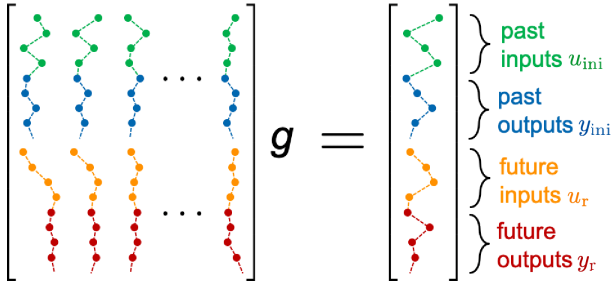


FIGURE 11 Schematic illustration of the DeePC formulation (45) as a regression problem. The vector g robustly selects elements from the trajectory library $\text{span}(\mathcal{H}_{T_{ini}+T_f}(w_d))$ to match the past pre-fix trajectory (u_{ini}, y_{ini}) as well as the future reference trajectory (u_f, y_f) . Robustness is enforced by choosing the regularizer in (45).

which extent the inner pre-processing/identification problems are (approximately) enforced. However, unlike in the indirect formulations, no projection on the 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 trade-off discussions give an intuition when indirect data-driven control approaches are inferior (respectively, superior) to the direct DeePC formulation; see [54], [65] for further elaborations.

Regularizations derived from robust optimization: One may

also take the robust minmax regression perspective presented in (S48) to robustify the certainty-equivalence LQ problem (42) as

minimize over g maximize over $\hat{w}_d \in \mathcal{W}(w_d)$

$$\|\mathcal{H}_{T_{ini}+T_f}(\hat{w}_d)g - w_{r,ini}\|_P^2,$$

where $\mathcal{W}(w_d)$ is an uncertainty set typically centered at the collected offline data w_d . Similar to the triangle inequality manipulation following (S48), one can show equivalence of 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 column-wise uncertainties, to uncertainties with Hankel structure. For each of these [64], [66] propose tractable reformulations, many of which result in regularization terms $h(g)$ taking the form of weighted p -norms: e.g., a uniform column-wise uncertainty (e.g., 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 the Sidebar 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 (42) have been proposed. We mention, among others, [68], [69] presenting a data-driven formulation of system-level and input-output parameterizations

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-square regression akin to (28) or (45).

Consider a set of n data pairs $\{x_i, y_i\} \in \mathbb{R}^p \times \mathbb{R}$, $i \in \{1, \dots, n\}$, denoting independent and dependent variables, i.e., 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 $Y = [y_1 \dots y_n]^\top$.

If we assume that the output measurements are subject to additive unit-variance Gaussian noise, then a maximization of the log-likelihood function (i.e., the likelihood of observing the data given the parameters) leads to the classic estimation criterion of minimizing the residual sum of squares [S1]:

$$\text{minimize}_\theta \|Y - X\theta\|_2^2 \quad (\text{S46})$$

A unique least-square solution $\theta^* = (X^\top X)^{-1} X^\top Y$ exists provided that X has full column rank. If that is not the case, then a standard modification to the optimization objective (S46) is Tikhonov (or ridge) regularization strongly convexifying the optimization problem (S46) by adding a quadratic penalty [S2]:

$$\text{minimize}_\theta \|Y - X\theta\|_2^2 + \lambda \|\theta\|_2^2 \quad (\text{S47})$$

For a positive coefficient $\lambda > 0$, a unique solution to (S47) of the form $\theta^* = (X^\top X + \lambda I)^{-1} X^\top Y$ always exists. Here, regularization acts as 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 below. They are both motivated from the fact that in case of noisy data the solution of the least-square problem (S46) might be overfitting the data set, and its solution θ^* may perform poorly out-of-sample, that is, when applying the estimated model $y = x^\top \theta^*$ to unseen data. Hence, either side information (prior knowledge) must be incorporated or the estimation criterion has to be robustified.

In a Bayesian setting, the parameter vector θ is explicitly regarded as a random variable, and a prior belief might be that all parameters θ_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 (S47) [S1]. In words, conditioning the regression (S46) 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 $Y = X\theta$, i.e., X is noise-corrupted. Then a robust version of the least-squares problem (S46) is

$$\min_\theta \max_{\Delta: \|\Delta\|_F \leq \rho} \|Y - (X + \Delta)\theta\|_2 \quad (\text{S48})$$

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

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

where the last equality follows from submultiplicativity. Finally, note that the triangle inequality is tight within the considered uncertainty set. Thus, the robust least-squares problem (S48) 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 ℓ_1 -regularizer $\|\theta\|_1$) can be derived from either Bayesian perspectives [S1] or (distributionally) robust problem formulations [S3], [S4].

In summary, regularizations (i) aid the optimization by rendering the solution unique, (ii) condition models on prior knowledge, and (iii) robustify problem formulations. These conceptual insights apply beyond the considered regression problem, e.g., in system identification [S5] [9] or in data-driven control [24], [56], [63], [64], as elaborated upon in the main text.

REFERENCES

- [S1] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, "The elements of statistical learning: data mining, inference, and prediction," in *Springer*, vol. 2, 2009.
- [S2] D. Bertsekas, "Convex optimization algorithms," in *Athena Scientific*, 2015.
- [S3] D. Bertsimas and M. S. Copenhaver, "Characterization of the equivalence of robustification and regularization in linear and matrix regression," in *European Journal of Operational Research*, vol. 270, no. 3, pp. 931–942, 2018.
- [S4] S. Shafieezadeh-Abadeh, D. Kuhn, and P. M. Esfahani, "Regularization via mass transportation," in *Journal of Machine Learning Research*, vol. 20, no. 103, pp. 1–68, 2019.
- [S5] G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, and L. Ljung, "Regularized System Identification: Learning Dynamic Models from Data," in *Springer*, 2022.

[70], [71] by means of the representation (16). Further, [72] directly robustifies the problem formulation (42) to adversarial and measurable disturbances. Finally, [73] proposes a maximum likelihood approach to DeePC. We refer to [16] for a detailed overview discussing and comparing different approaches.

DeePC: closed-loop & implementation aspects

So far we discussed DeePC (43) 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 the Sidebar Selected Implementations of DeePC.

Stability: First and foremost, we discuss closed-loop stability and robustness in presence of noisy data. The sequence of articles [26], [75]–[78] has shown that many stability and robustness certificates for MPC carry over to the data-driven domain, but particular care must be taken when addressing, e.g., noise in the data matrix $\mathcal{H}_{T_{\text{ini}}+T_f}(w_d)$ and terminal ingredients (needed for

DISTRIBUTIONAL ROBUSTNESS IN TRAJECTORY SPACE

In the regression formulation of DeePC (45), the columns (or rows) of the Hankel matrix $\mathcal{H}_{T_{ini}+T_f}(\hat{w}_d)$ are trajectories, and the objective is to minimize the loss by linearly combining them; see Figure 11. We now take a stochastic perspective and consider each trajectory as an observed sample from some unknown stochastic process. Then we can equivalently reformulate problem (45) (without regularization term) as

$$\text{minimize over } g \quad \mathbb{E}_{\hat{w}_d \sim \hat{\mathbb{P}}} \left[\left\| \mathcal{H}_{T_{ini}+T_f}(\hat{w}_d)g - w_{r,ini} \right\|_P^2 \right], \quad (\text{S51})$$

where $\hat{\mathbb{P}}$ is the associated empirical distribution built using the data samples w_d , i.e., the measure of \hat{w}_d is supported on w_d .

When implementing the solution of the so-called sample average approximation (S51) 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, [24], [74] propose the *distributionally robust formulation*

$$\inf_{g \in \mathcal{G}} \sup_{Q \in \mathcal{B}_\varepsilon(\hat{\mathbb{P}})} \mathbb{E}_{\hat{w}_d \sim Q} \left[\left\| \mathcal{H}_{T_{ini}+T_f}(\hat{w}_d)g - w_{r,ini} \right\|_P^2 \right], \quad (\text{S52})$$

where the expectation is taken with respect to an unknown distribution Q taking value in a so-called *ambiguity set* of all distributions which are ε -close to the sample distribution $\hat{\mathbb{P}}$. Problem (S52) seeks to be robust to all such distributions 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 $\mathcal{B}_\varepsilon(\hat{\mathbb{P}})$. Let us elaborate on these ingredients from a high-level and accessible perspective.

More formally, let $\mathcal{M}(\Xi)$ be the set of all distributions Q supported on Ξ (a subset of a vector space) such that $\mathbb{E}_Q[\|\xi\|_p] < \infty$, where $\|\cdot\|_p$ is the p -norm for some $p \in [1, \infty]$. The *Wasserstein distance* $d_W: \mathcal{M}(\Xi) \times \mathcal{M}(\Xi) \rightarrow \mathbb{R}_{\geq 0}$ between two distributions Q_1 and Q_2 , each supported on Ξ , is

$$d_W(Q_1, Q_2) = \inf_{\Pi} \left\{ \int_{\Xi} \int_{\Xi} \|\xi_1 - \xi_2\|_p d\Pi(\xi_1, d\xi_2) \right\}, \quad (\text{S53})$$

where Π is the set of distributions over $\mathcal{M}(\Xi \times \Xi)$ with marginal distributions $Q_1 \in \mathcal{M}(\Xi)$ and $Q_2 \in \mathcal{M}(\Xi)$, respectively.

The semantics of (S53) are as follows: we seek the minimum expected cost to transport the probability distribution Q_1 onto the probability distribution Q_2 when transporting a unit of mass from ξ_1 to ξ_2 costs $\|\xi_1 - \xi_2\|_p$. We refer to Figure S1 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 Π that map Q_1 onto Q_2 , the Wasserstein distance seeks the minimum cost solution.

Finally, given the distance (S53), we denote the *Wasserstein ball* of radius $\varepsilon \geq 0$ centered around a distribution Q by

$$\mathcal{B}_\varepsilon(Q) = \{Q' \in \mathcal{M}(\Xi) \mid d_W(Q, Q') \leq \varepsilon\}.$$

The Wasserstein distance has intimate connection to optimal transport [S1], [S2], and unlike other distances (or 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 distributional robust optimization problem as (S53) [S3], [S4].

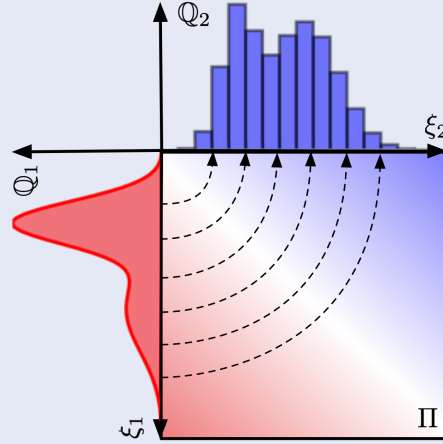


FIGURE S1 Schematic illustration of the Wasserstein distance between two distributions. Here Q_1 is an absolutely continuous distribution, Q_2 is an empirical distribution, and Π is a joint distribution (or a transport plan) with marginals Q_1 and Q_2 .

To return to problem (S52), it ensures robustness to all time series which are ε -close to the data samples \hat{w}_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, [S5] 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 (S52), [24] has shown that, when the objective admits a Lipschitz constant L , (S52) is equivalent to a version of regularized DeePC (44),

$$\text{minimize over } g \quad \left\| \mathcal{H}_{T_{ini}+T_f}(\hat{w}_d)g - w_{r,ini} \right\|_P + L\varepsilon \cdot \|g\|_q$$

where $1/p + 1/q = 1$, i.e., $\|\cdot\|_q$ is the dual norm of the one used to construct the Wasserstein ball. For example, safeguarding against uncertainty in ℓ_∞ -norm in the space of trajectories is equivalent to ℓ_1 -norm regularization. The radius ε of the ambiguity can be estimated (albeit conservatively) from the number of data samples, and it can be reduced by averaging data sets.

REFERENCES

- [S1] C. Villani, "Optimal transport: old and new," in Springer, vol. 338, 2009.
- [S2] G. Peyré and M. Cuturi, "Computational optimal transport: With applications to data science," in Foundations and Trends in Machine Learning, vol. 11(5-6), pp. 355–607, 2019.
- [S3] D. Kuhn, P. Esfahani, V. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," in Operations Research & Management Science in the Age of Analytics, INFORMS, 2019, pp. 130–166.
- [S4] J. Blanchet, J. Karthyek Murthy, and V. Anh Nguyen, Statistical analysis of Wasserstein distributionally robust estimators," in Tutorials in Operations Research: Emerging Optimization Methods and Modeling Techniques with Applications. INFORMS, 2021. 227–254.
- [S5] S. Shafieezadeh-Abadeh, D. Kuhn, and P. M. Esfahani, "Regularization via mass transportation," in Journal of Machine Learning Research, vol. 20, no. 103, pp. 1–68, 2019.

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 [52], an autonomous walking excavator [S1], leading cruise control [S2], and a quadruped [S3] next to other small-scale laboratory implementations; see [16]. Within the energy domain, DeePC has been implemented on grid-connected power converters [66], synchronous motor drives [S4], [S5], and building automation [S6], [S7]. Further, [72] provides a decentralized DeePC implementation for power system oscillation damping, and [S8] 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. The Section “Application to Control of Power Electronics Dominated Power Systems” will present further case studies.

REFERENCES

[S1] F. Wegner, J. Coulson, M. Hudoba de Badyn, J. Lygeros, and S. Trimpe, “Data-enabled predictive control of a 12t excavator,”

ETH Zurich, Master Thesis, 2021, available at <https://www.research-collection.ethz.ch/>.

[S2] J. Wang, Y. Zheng, J. Dong, C. Chen, M. Cai, K. Li, and Q. Xu, “Experimental validation of deep-lcc for dissipating stop-and-go waves in mixed traffic,” *arXiv preprint arXiv:2204.03747*, 2022.

[S3] R. T. Fawcett, K. Afsari, A. D. Ames, and K. A. Hamed, “Toward a data-driven template model for quadrupedal locomotion,” *IEEE Robotics and Automation Letters*, 2022.

[S4] P. Carlet, A. Favato, S. Bolognani, and F. Dörfler, “Data-driven predictive current control for synchronous motor drives,” in *2020 IEEE Energy Conversion Congress and Exposition*, 2020, pp. 5148–5154.

[S5] P. Carlet, F. A., S. Bolognani, and F. Dörfler, “Data-driven continuous-set predictive current control for synchronous motor drives,” *IEEE Transactions on Power Electronics*, 2021, submitted.

[S6] 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 preprint arXiv:2106.05740*.

[S7] J. Schwarz, F. Micheli, M. Hudoba de Badyn, and R. Smith, “Data-driven control of buildings and energy hubs,” ETH Zurich, Semester Thesis, 2019, available at <https://www.research-collection.ethz.ch/>.

[S8] P. Mahdavi-pour, C. Wieland, and H. Spliethoff, “Optimal control of combined-cycle power plants: A data-enabled predictive control perspective,” in *IFAC Conference on Networked Systems*, July 2022, pp. 103–106.

stability) which are normally computed from model information. The initial work [67] certifies closed-loop practical exponential stability of DeePC (43) with input constraints and when applied to noisy LTI systems. Some salient features of the analysis methods include a ridge regularizer $h(g) = \|g\|_2^2$ and terminal equilibrium constraints. Various extensions have been brought forward such as terminal costs and constraints computable from data [76] as well as implementations without terminal ingredients [77]. Further modifications address robust output constraint satisfaction [78], time-varying references [75], and linear tracking control for affine and nonlinear systems [26].

Nonlinearity: To remain on the theme of nonlinearity, the DeePC methods is founded on LTI systems: an optimal trajectory is synthesized by linearly (albeit robustly) combining trajectories from a library according to the superposition principle. Though, many case studies (see the Sidebar Selected Implementations of DeePC) show that DeePC performs remarkably well in closed loop with weakly nonlinear systems and often outperforms approaches based on linear SysID. To provide a first explanation of 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 [54] is that the DeePC formulation (43) 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

[16] for a review. In either case, so far only [26] has provided theoretic certificates when applying DeePC (with a sliding data window) in closed loop with a slowly varying nonlinear system.

Constraints & feasibility: We have already touched upon terminal constraints ensuring closed-loop stability. We now revisit the other constraints in (43). The estimation constraints, $U_{pg} = u_{ini}$ and $Y_{pg} = y_{ini}$, have been softened within DeePC (43) by means of slack variables. Instead of such a finite horizon estimation, one may also adopt a Kalman filtering approach [79] to recursively construct an estimate of a hidden state. Regarding output constraints, $Y_{rg} \in \mathcal{Y}$, 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 [24]. Finally, [78] provides closed-loop stability and feasibility certificates for tightened constraints.

Trajectory library: As discussed previously, the data matrix $\mathcal{H}_{T_{ini}+T_f}(w_d)$ 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 less 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 is conditioned on time-invariance (for Hankel matrices) or columns are independent (for trajectory or Page matrices). The former narrows the hypothesis space for optimal trajectories, which is advantageous in case of

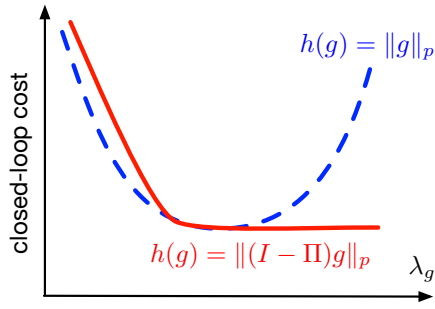


FIGURE 13 Qualitative illustration of the realized closed-loop performance as a function of the coefficient λ_g for a projection-based (red solid) and a norm-based (blue dashed) regularizer. This qualitative behavior has been observed by the authors across many case studies involving LTI systems and weakly nonlinear systems.

inexact data. The latter allows for less structured pre-processing (e.g., low-rank approximation by singular value thresholding) or robustification, e.g., columnwise uncertainties can be considered resulting in weighted ℓ_1 -regularizations. Finally, empirical results often show a superior realized performance when choosing less structured trajectory or Page matrices [24], [64], [72].

Computational implementation: To efficiently implement DeePC, the OSQP solver [80] proved to be a formidable choice. A comparison of the compactified DeePC formulation (43) and the model-based predictor (37) shows that DeePC contains more decision variables (even when eliminating some as in (44)), 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 (S41), or multiple shooting formulations [81]. 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, de-noised, and tidied-up representations.

Tuning of hyper-parameters: We highlight the simplicity of the DeePC implementation: as in MPC and MHE a cost function, estimation horizon T_{ini} , prediction horizon T_f , and terminal ingredients need to be chosen; see [43], [44] 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 which have proved useful in many case studies, and we refer to the next section for explicit examples.

In terms of data collection, the length T_d of the data time series w_d has to be sufficiently long to assure persistency of excitation; see the fundamental lemma. The lower bound for T 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_d typically suffices, and the authors often had good experiences with choosing T_d so that the data matrix $\mathcal{H}_{T_{\text{ini}}+T_f}(w_d)$ is square.

For the estimation penalty a quadratic cost, $\lambda_{u_{\text{ini}}} \|\sigma_{u_{\text{ini}}}\|_2^2 + \lambda_{y_{\text{ini}}} \|\sigma_{y_{\text{ini}}}\|_2^2$, with sufficiently large coefficients ($\lambda_{u_{\text{ini}}}, \lambda_{y_{\text{ini}}}$) has

proved useful in terms of performance and computational efficiency. As for the choice of regularizers and associated coefficients, the projection-based regularizer (50) with sufficiently large coefficient λ_g has proved useful, especially for LTI systems where it does not induce any bias. Often a squared regularizer is preferred by quadratic programming solvers. Norm-based regularizers $h(g) = \|g\|_p$ may perform equally well but are harder to tune: the realized performance improves when increasing λ_g beyond a certain threshold, remains constant for a large interval of λ_g , and then increases again beyond a second threshold; see Figure 13. The (fortunately often shallow) minimum has to be found by cross-validation or deduced from prior knowledge (e.g., from the size of an uncertainty set). As a rule of thumb, an optimal λ_g can be practically found by logarithmically increasing λ_g . The case study in the next section will revisit this discussion.

Summary: This section has presented different indirect (i.e., model-based) and direct data-driven control methods. The DeePC formulation (43) 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 system identification 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 to achieve *net zero*. As a dominant carbon emitter, the power and energy industry plays a vital role towards this goal, where integrating renewable energy sources is one of the primary approaches for decarbonization.

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 non-dispatchable [82], [83]. 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 direct current (DC) circuits with alternating current (AC) circuits [84]. 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 14. 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 and more fossil-fuel-based power plants replaced by renewable energy sources, many countries worldwide are currently witnessing the establishment of *power electronics dominated power systems* in the near future, as shown in Figure 14.

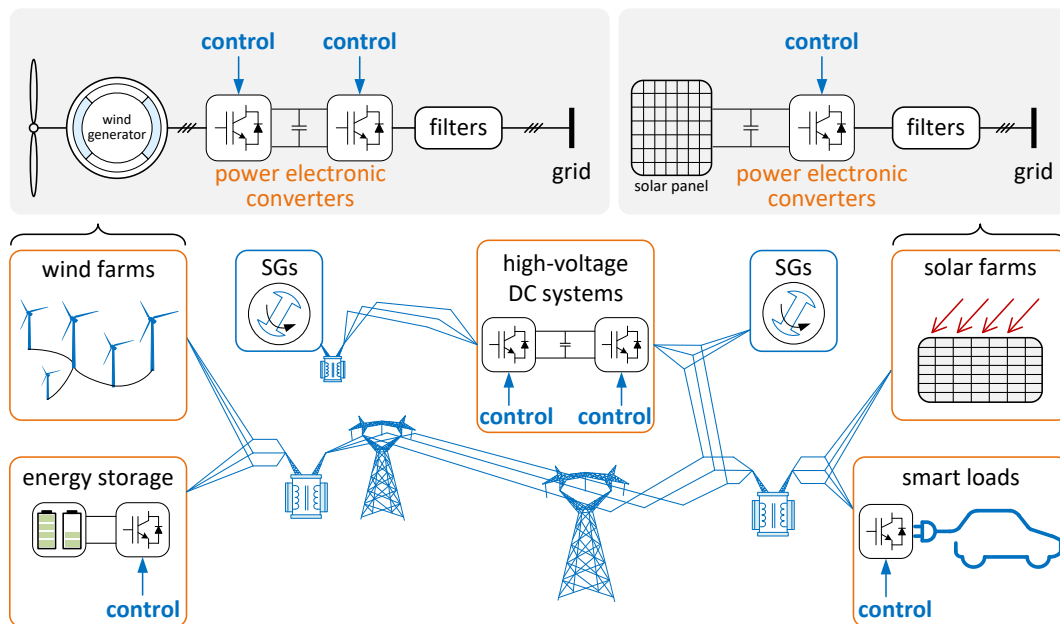


FIGURE 14 An illustration of a power electronics dominated power system.

Unlike SGs, power converters have fast actuation and high flexibility in controlling the voltage angle and magnitude. For instance, according to the Newton's laws of motion, one can only indirectly change the rotor angle of a 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 the Sidebar Challenges in Control of Power Electronics Dominated Power Systems.

The dynamics of a grid-connected power converter depends on both the inherent control scheme as well as the 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 [53]. On the other hand, power converter supplier usually do not share their proprietary models to the manufacturers of wind generators. Moreover, in a wind farm, there are complex aerodynamic interactions among the wind turbines, which are in general difficult to model and treated as disturbances [85]. 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 the above question. We consider both device-level control (including two scenarios: 1) stabilization of wind generators as well as 2) 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 I: 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 the Sidebar 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 [86], [87]. 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,

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, high-voltage DC transmission systems, etc. In these devices, by performing pulse-width 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 [S1].

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 time scales, but to maintain desired performance, the controller should be robust against a variety of uncertainties. On the other hand, in a very fast time scale (milliseconds), the controller needs to be designed and implemented in a fully decentralized manner considering constraints (e.g., 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, the above requirements are handled by employing multiple nested control loops based on PID controllers to achieve power/voltage regulations and grid synchronization in power converters, and the constraints are implemented using saturations and anti-windup strategies [S2], [S3]. The corresponding control structure (i.e., 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: (i) often the power converter manufacturers do not share detailed proprietary models to the control designer of wind turbine manufacturers, and vice versa; (ii) 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; and (iii) finally exogenous disturbances (e.g., intermittency of renewables) are hard to model, especially on fast time scales. 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, e.g., those in the U.S. and China, especially when the wind farms are integrated in weak power grids [S4]. 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 is readily available during the operation of a power system, which can be used to capture the system dynamics and perform optimal control [72], [S5]. In the Section Case study I: stabilizing wind generators, we demonstrate how direct data-driven control excels in stabilizing wind generators by performing optimal control.

REFERENCES

- [S1] F. Blaabjerg, R. Teodorescu, M. Liserre, and A.V. Timbus, "Overview of control and grid synchronization for distributed power generation systems," in *IEEE Transactions on industrial electronics*, vol. 53, no. 5, pp. 1398-1409, 2016.
- [S2] L. Huang, H. Xin, and F. Dörfler, "H ∞ -control of grid-connected converters: Design, objectives and decentralized stability certificates," in *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 3805-3816, 2020.
- [S3] L. Huang, H. Xin, Z. Wang *et al.*, "Transient stability analysis and control design of droop-controlled voltage source converters considering current limitation," in *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 578-591, 2019.
- [S4] Y. Cheng, L. Fan, J. Rose *et al.*, "Wind energy systems sub-synchronous oscillations: Events and modeling," IEEE Power and Energy Society, Technical Report, PES-TR80, 2020.
- [S5] A. Chakraborty, "Wide-area damping control of power systems using dynamic clustering and TCSC-based redesigns," in *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1503-1514, 2012.

to illustrate the point we consider the direct-drive (type 4) wind generator in Figure 15. 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, maximum power point tracking, etc. It has 37 state variables and is highly nonlinear due to the nonlinear power flow equation, maximum power point tracking curve, (adaptive) phase-locked loop (PLL), etc. In industrial practice

this model and its parameters would be proprietary and unknown to commissioning control engineers and system operators of the power grid. It might also be difficult to use conventional system identification 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 (i.e., 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 single-device system, the short-circuit ratio is the inverse of the grid impedance (in per-unit

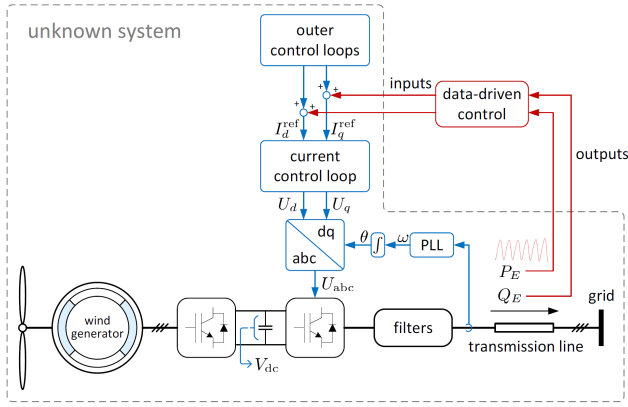


FIGURE 15 Application of data-driven control to stabilize a direct-drive wind generator.

values), which characterizes the strength of the power grid [88]. The studied wind generator relies on a synchronous reference frame PLL [89], [90] 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 60rad/s to avoid strongly time-varying dynamics. When connected to such a weak power grid, the wind generator has 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 15, 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 added to the current references. The input/output signals are all in per-unit values (p.u.). 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: $T_d = 600$, $T_{ini} = 6$, $T_f = 12$, $R = I$, $Q = 100I$, and $\lambda_{y_{ini}} = 1000$. 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, respectively. Note that the hyper-parameters in our case studies are selected manually, and one may achieve better performance by finding a better set of parameters; see Figures 17, 19 later on. The control horizon is $k = 1$, i.e., we apply the first k elements of u_f to the system before solving DeePC again, see Figure 10. The reference trajectory is $r = \mathbf{1} \otimes \begin{bmatrix} P_s \\ Q_s \end{bmatrix}$, 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 5ms. 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; the lower and upper bounds of the reactive

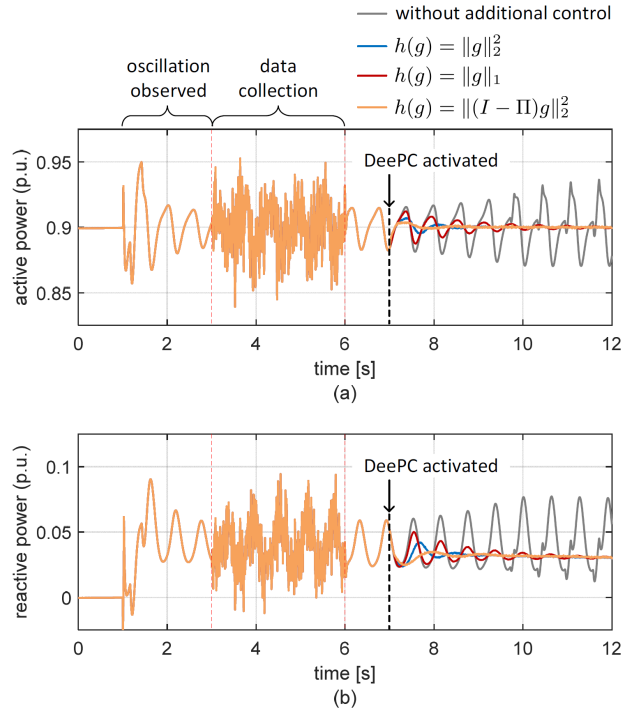


FIGURE 16 Time-domain responses of the wind generator: (a) active power and (b) reactive power.

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 are used [66].

Figure 16 shows the time-domain responses of the wind generator. At $time = 1s$, we change the short-circuit ratio from 2.23 to 2 to emulate an event in the power grid, e.g., 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 [86]. 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 5ms, we only require 3s (from $time = 3s$ to $time = 6s$ as displayed in Figure 16) 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, e.g., a pseudorandom binary sequence. Moreover, one may also avoid actively injecting excitation signals into the system by collecting long enough oscillation data that satisfies the persistency of excitation condition. The output signals are subject to measurement noise with the signal-to-noise ratio (SNR) being 2×10^3 . Here the SNR is defined as $SNR = \|\Delta y_d\|_2^2 / \|y_{noise}\|_2^2$, where Δy_d is the deviation of y_d from its steady-state value and y_{noise} is the noise sequence; it reflects how the noise compares

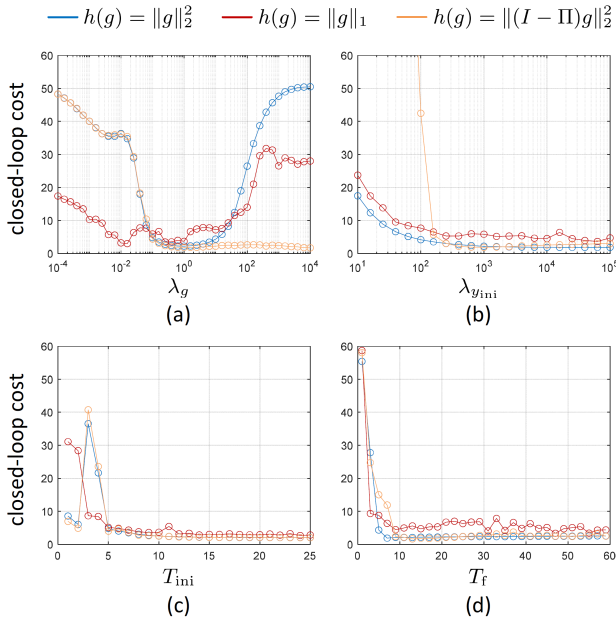


FIGURE 17 Averaged realized closed-loop cost of the system (a) with different values of λ_g (b) with different values of $\lambda_{y_{ini}}$ (c) with different values of T_{ini} , and (d) with different values of T_f .

to the fluctuation in the outputs caused by the excitation signals.

The DeePC method is activated at $time = 7s$. 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 [80] 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 16GB RAM, it takes about 0.013s to solve DeePC with the quadratic regularizer; it takes about 0.02s to solve DeePC with the one-norm regularizer; it takes about 0.02s 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 $\sum_t \|u_t\|_R^2 + \|y_t - r_t\|_Q^2$ measured from the system from 7s to 12s. Figure 17 (a) shows the averaged closed-loop cost of the system when different regularization coefficients λ_g are used. The averaged closed-loop cost is computed by repeating the simulation for 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 λ_g . As mentioned before, the regularization on g in the

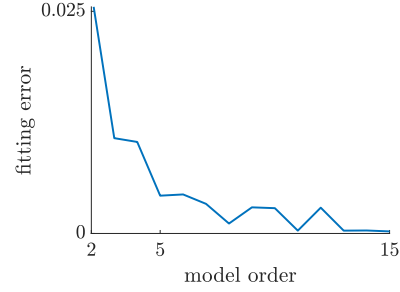


FIGURE 18 Fitting error when assuming different model orders to identify the system.

cost function assures robustness. Hence, the regularizers achieve satisfactory performance when λ_g is large enough. However, with the quadratic regularizer or the one-norm regularizer, the performance deteriorates with a too large λ_g , as it conservatively hedges against a too large inherent uncertainty set that results in poor performance of the obtained control sequence [64]. By comparison, the projection-based regularizer still maintains excellent performance even with a large λ_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 λ_g is applied. In summary, Figure 17 (a) quantitatively confirms the illustration in Figure 13.

Figure 17 (b) shows the averaged closed-loop cost with different values of $\lambda_{y_{ini}}$, which indicates that a sufficiently large $\lambda_{y_{ini}}$ is needed to ensure satisfactory performance; see also [64, Theorem III.1]. Figure 17 (c) and (d) show the averaged closed-loop cost with different values of T_{ini} and T_f , respectively. It can be seen that the system achieves excellent performance when T_{ini} and T_f are sufficiently large, consistent with the results in [52], [53], [72], etc. Usually the prediction horizon should be large enough to ensure stability of the system [91]. We notice that the lag of the system ℓ is at least 19 since the system has 37 state variable and 2 outputs, and theoretically T_{ini} should be larger than ℓ . However, we observe from Figure 17 (c) that the closed-loop system shows satisfactory performance with $T_{ini} \geq 5$. This may be because the data mainly captures 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 18, which shows that the optimal model order is $n = 5$ (in terms of balancing the fitting error and the complexity). Hence, aligned with insights from model-order reduction, in practice one may expect good performance even when T_{ini} is smaller than the theoretical lower bound inferred from the real system order.

Figure 19 (a) plots the averaged closed-loop cost with different values of T_d , which shows that a sufficiently large T_d leads to satisfactory performance, and increasing T_d does not always improve the performance. For instance, with the projection-based regularizer, we observe the best performance when T_d is around

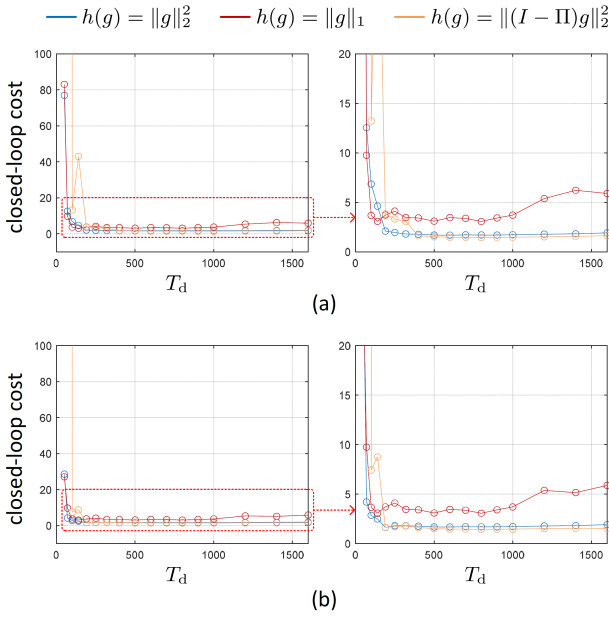


FIGURE 19 Averaged realized closed-loop cost of the system with different length T_d of input/output trajectory to construct Hankel matrices (a) with a fixed λ_g and (b) for each value of T_d , the optimal value of λ_g is chosen from $\{0.001, 0.01, 0.1, 1, 10, 100\}$.

800; with the one-norm regularizer, the performance starts to degrade when $T_d \geq 1000$. According to the interpretation from min-max optimization (See Sidebar Roles of Regularization), one may need to choose a larger λ_g when T_d increases. Figure 19 (b) further shows the averaged closed-loop cost where for each value of T_d , the optimal value of λ_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 19 (a). Moreover, we observe that the optimal value of λ_g increases with a larger T_d (data not shown). Overall, all the regularizers have satisfactory performance when T_d is sufficiently large.

In short, the above 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_d , $\lambda_{y_{ini}}$, T_{ini} , and T_f .

Case study II: 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 activated only when oscillations/instabilities are observed. During normal operations, the wind turbines mainly use PID controllers to achieve multiple control objectives, e.g., 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 ω ; see Figure 20 (a). The PLL aims to regulate V_q to be 0 at 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

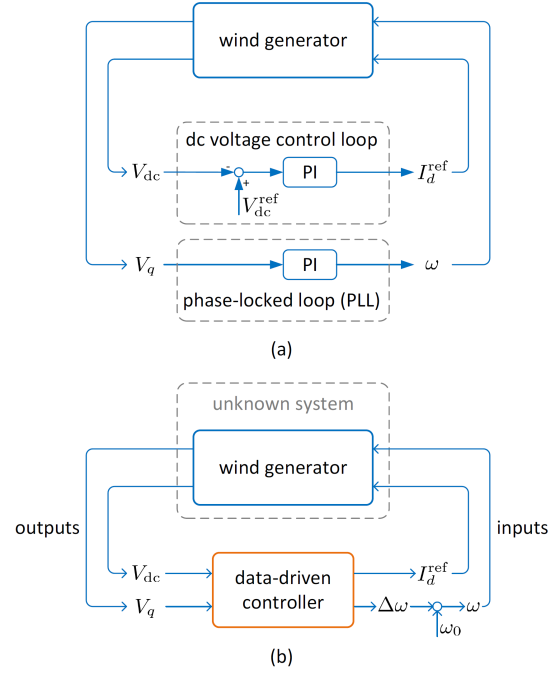


FIGURE 20 DC voltage control and grid synchronization control of wind generators (a) using PI controllers and (b) using data-driven control.

error (i.e., $V_{dc}^{ref} - V_{dc}$) to generate the d-axis (active) current reference I_d^{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 20 (b), we use the data-driven controller to replace the conventional DC voltage control loop and PLL, which measures the system outputs V_{dc} and V_q and provides control inputs I_d^{ref} and $\Delta\omega$ (the deviation of ω from the nominal frequency ω_0) to the system. In this manner, the converter's frequency is generated using both DC voltage and q-axis voltage as feedback signals. In fact, some *grid-forming converters* also use the DC voltage to generate the frequency, e.g., those in [92], [93]. 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 [94]–[98]. The data-driven controller in Figure 20 (b) has the potential to become a general optimal control framework that covers both grid-forming control and PLL-based (i.e., 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 per-unit values, and the parameters of DeePC are: $T_d = 800$, $T_{ini} = 5$, $T_f = 25$, $R = I \otimes \text{diag}(1, 20)$, $Q = 1000I$, and $\lambda_{y_{ini}} = 1000$. We choose $\lambda_g = 1$

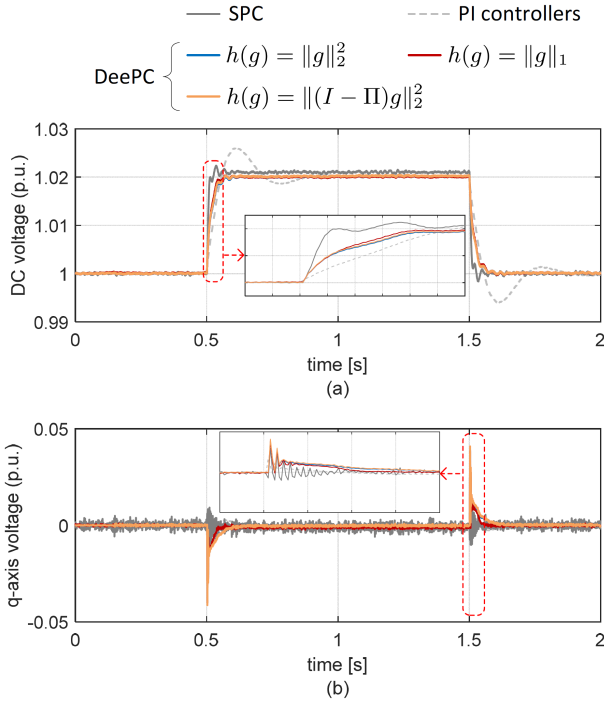


FIGURE 21 Time-domain responses of the wind generator: (a) DC voltage and (b) q-axis voltage using PI control, SPC, and DeePC.

for the quadratic regularizer and the projection-based regularizer, and $\lambda_g = 0.01$ for the one-norm regularizer. The control horizon is $k = 1$. The reference trajectory is $r = \mathbf{1} \otimes \begin{bmatrix} V_{dc}^{ref} \\ 0 \end{bmatrix}$. 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 1ms to handle fast DC-link dynamics. In the constraints, the lower and upper bounds of I_d^{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 V_{dc} (output 1) are 0.9 and 1.1 , respectively; the lower and upper bounds of V_q (output 2) are -0.2 and 0.2 , respectively.

Before DeePC is activated, we inject white noise signals for 0.8s to excite the system and collect input/output data. During this period, I_d^{ref} comes from the sum of a white noise signal (variance: 4×10^{-3}) and a DC voltage control loop (shown in Figure 20 (a)) which pre-stabilize the DC voltage; ω comes from the sum of a white noise signal (variance: 1.75×10^{-5}) and a PLL which pre-synchronize 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 21 shows the time-domain responses of the wind generator, where the DC voltage reference steps from 1p.u. to 1.02p.u. at 0.5s, and steps back to 1p.u. at 1.5s. 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 (S41), which

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

averaged closed-loop cost	SNR: 10^4	SNR: 10^2
Hankel matrices ($h(g) = \ g\ _2^2$)	17.12	25.34
Page matrices ($h(g) = \ g\ _2^2$)	16.50	24.49
Hankel matrices ($h(g) = \ g\ _1$)	31.03	49.60
Page matrices ($h(g) = \ g\ _1$)	24.38	35.54
Hankel matrices ($h(g) = \ (I - \Pi)g\ _2^2$)	40.73	35.48
Page matrices ($h(g) = \ (I - \Pi)g\ _2^2$)	29.23	28.33
Hankel matrices (SPC)	292.07 unstable (1 case)	187.12 unstable (1 case)
Page matrices (SPC)	1.15×10^3 unstable (3 cases)	1.35×10^4 unstable (1 case)

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).

Table 1 shows the effects of choosing Page matrices instead of Hankel matrices in DeePC. Each of the simulations are repeated 100 times with different random seeds to generate the excitation signals and the measurement noise. 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 ($T_d = 23130$) 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. By comparison, all the simulation cases are stable when DeePC is applied.

Case study III: frequency control provided by wind farms

In what follows, we apply DeePC in a wind farm to perform optimal frequency control and improve the frequency stability of power systems (i.e., reduce frequency deviations and increase the frequency nadir). Conventionally, wind generators operate in maximum power point tracking 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 [82], [99]. 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 [100], [101] without operating the wind generators in de-loading 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 (IEEE

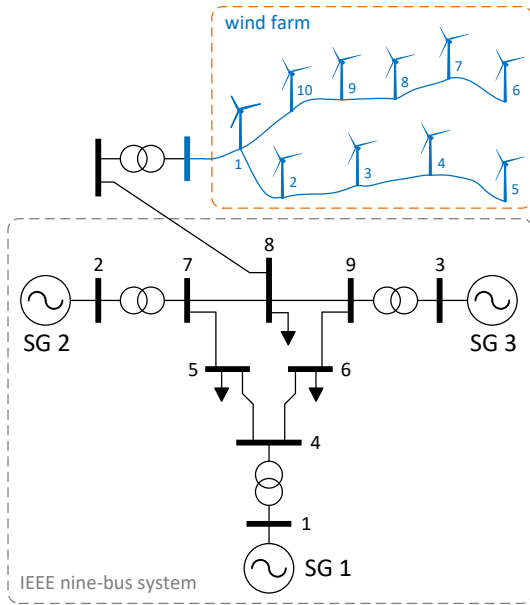


FIGURE 22 A wind farm connected to a power grid (IEEE nine-bus system). The wind speed at wind generators 1 ~ 5 is 10m/s, and the wind speed at wind generators 6 ~ 10 is 9.5m/s due to the wake effect. The total active power output of the three synchronous generators (SGs) and the wind farm are 300MW and 51.5MW, respectively.

nine-bus system), where the wind farm consists of 10 direct-drive wind generators (same model as the previous case studies), as shown in Figure 22. The models of the power grid and the wind farm are unknown from a control design perspective. We consider the following DeePC setting: 1) the output of the unknown system is the deviation of the frequency of SG 1 from the nominal frequency (Hz), with the lower and upper bounds being ± 0.5 ; this output signal can be obtained from wide-area measurements; 2) for each wind generator, DeePC provides an auxiliary control input signal added to the rotor speed reference (generated from the maximum power point tracking scheme) in the control of the generator-side converter, with the lower and upper bounds being ± 0.3 ; 3) the parameters of DeePC are: $T_d = 500$, $T_{ini} = 10$, $T_f = 25$, $R = I$, $Q = 40I$, $\lambda_{y_{ini}} = 40$, $k = 1$; we choose $\lambda_g = 2$ for the quadratic regularizer, and $\lambda_g = 0.4$ for the one-norm regularizer and the projection-based regularizer; 4) we choose the sampling time to be 0.1s, as we focus on slow power grid frequency dynamics.

During the data collection period, we simultaneously inject the same excitation sequence into all the wind generators to cause fluctuations in the system output (within 0.05Hz); the output is subject to measurement noise ($\text{SNR}: 5 \times 10^5$). 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 the wind generators, i.e., they operate together as a virtual power plant [102]. DeePC can also be implemented locally in each wind generator to avoid delays in the control

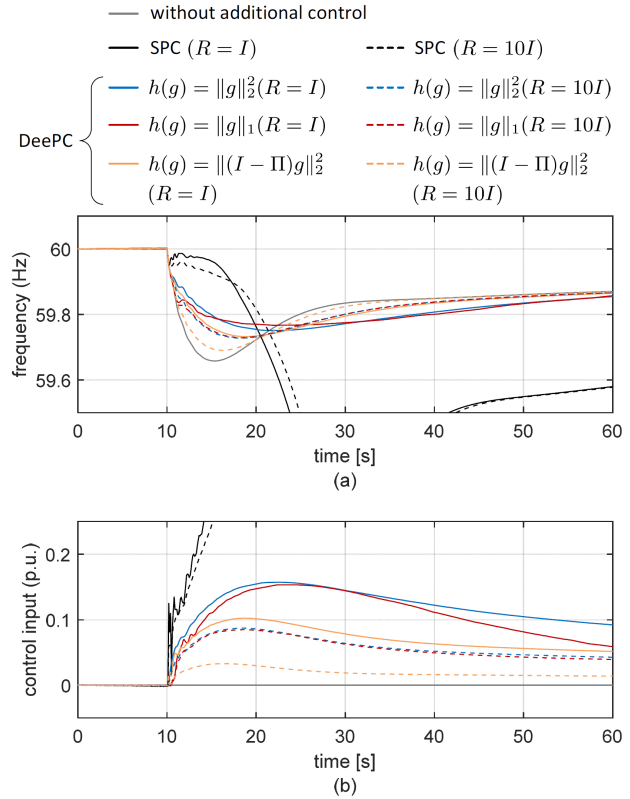


FIGURE 23 Time-domain responses of the system: (a) frequency of SG 1 and (b) control inputs provided by DeePC.

actuation; see [72] for an example. We will 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 23 shows the time-domain responses of the system when an overload event (bus 6) occurs at $time = 10s$ and causes a frequency dip. We employ DeePC in a central controller that provides identical control sequence to all the wind generators. It can be seen from Figure 23 (a) that the frequency nadir is improved when DeePC is applied, indicating better frequency stability. For instance, the frequency nadir is lifted from 59.66Hz to 59.75Hz 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 the cost of operating in de-loading mode, as we utilize the kinetic energy stored in the spinning turbine to support the grid frequency. Figure 23 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

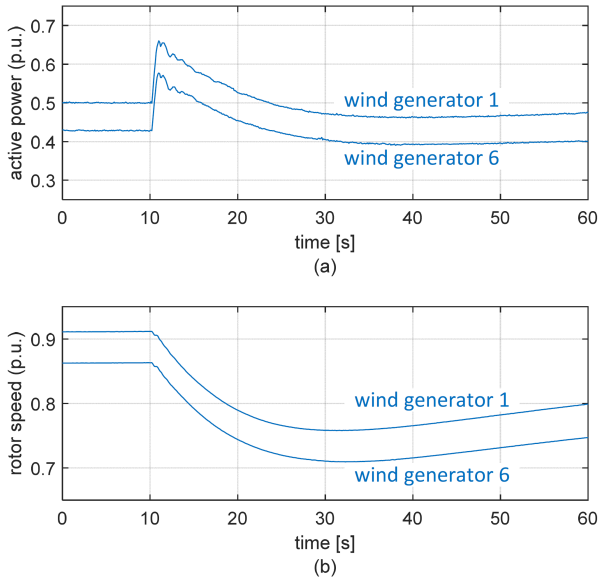


FIGURE 24 Time-domain responses of the wind generators with DeePC ($h(g) = \|g\|_2^2, R = I$): (a) active power and (b) rotor speed.

performance of SPC. However, as shown in Figure 23, the system becomes unstable in this case. We suspect that this is due to the nonlinearity of the system (e.g., the nonlinear maximum power point tracking curve and the nonlinear relationship between the captured wind power and rotor speed).

Figure 24 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 ~ 4 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 deviates from the maximum power point tracking curve during this process, and the rotor speeds will be restored to the original point to preserve the maximum power point tracking 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 the above simulations, it only takes about 0.02s to solve the optimization problem, and thus the algorithm can be implemented in real time.

Instead of directly using wide-area measurements, we next consider the scenario where each wind generator uses an additional PLL (bandwidth: 5rad/s) to locally detect the grid frequency and implement DeePC in a decentralized manner. Again, we simultaneously inject the same excitation sequence into all the wind generators during the data collection. However, different wind generators will observe different grid frequency responses through their own PLLs. Figure 25 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.

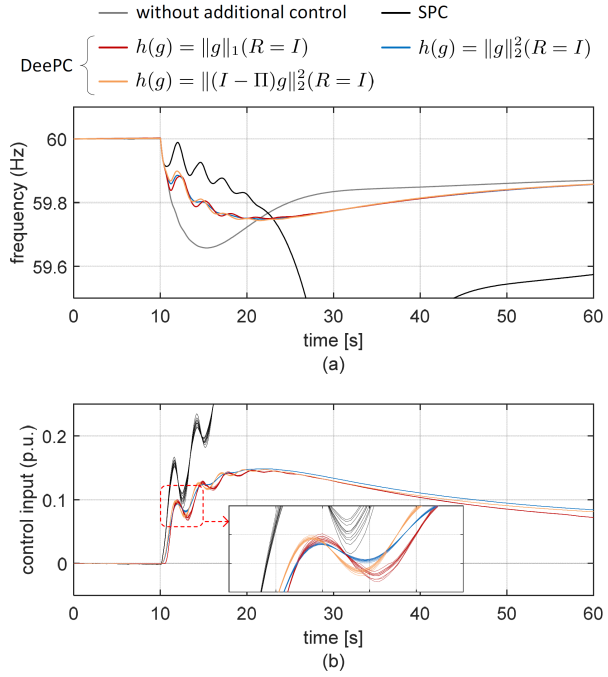


FIGURE 25 Time-domain responses when PLL measurements are used: (a) frequency of SG 1 and (b) control inputs of all the wind generators provided by DeePC in a decentralized manner.

Here 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 hyper-parameters 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 23, the frequency responses in Figure 25 have lower but still satisfactory damping ratio. Overall, the above 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) 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 of 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 behavioral paradigm defines a dynamical system as a set of trajectories, thus decoupling it from its parametric representations. For an exact trajectory of a linear time-invariant system, under the persistency of excitation condition (15), the image of a Hankel matrix constructed from the data coincides with the behavior of the system. This allows us to develop algorithms for

data-driven analysis, signal processing, and control.

We presented a generic data-driven interpolation/ approximation formulation encompassing many well known problems instances, among others finite-horizon data-driven control. We dwelled deeper into this problem, embedded it into a predictive control setting, and robustified it by means of regularizations. Finally, we presented applications of direct data-driven control in the context of power electronics dominated power systems.

Fruitful avenues for future research include a principled and bottom up extension of these methods to stochastic and nonlinear systems starting from their behaviors. We believe that real-time adaptation of the presented methods holds great value both for theory as well as practical implementations. For the latter we also highlight automatic tuning of hyper-parameters as well as computational 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 paper.

The research leading to these results received funding from: the Catalan Institution for Research and Advanced Studies (ICREA), the Fond for Scientific Research Vlaanderen (FWO) projects G090117N, G033822N, FNRS-FWO EOS project 30468160, European Union's Horizon 2020 research and innovation program (Grant Agreement Number 883985), Swiss National Science Foundation under NCCR Automation, and ETH Zurich Funds.

AUTHOR INFORMATION

Ivan Markovsky (imarkovsky@cimne.upc.edu) is an ICREA professor at the International Centre for Numerical Methods in Engineering, Barcelona. He received his 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 an ERC starting grant "Structured low-rank approximation: Theory, algorithms, and applications" 2010–2015, Householder Prize honorable mention 2008, and research mandate by the Vrije Universiteit Brussel research council 2012–2022. His main research interests are computational methods for system theory, identification, and data-driven control in the behavioral setting.

Linbin Huang (linhuang@ethz.ch) is a postdoctoral researcher with the Automatic Control Laboratory at ETH Zürich. 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 (dorfler@ethz.ch) is an Associate Professor at the Automatic Control Laboratory at ETH Zürich and the Associate Head of the Department of Information Technology and Electrical Engineering. He received his Ph.D. degree in

Mechanical Engineering from the University of California at Santa Barbara in 2013, and a 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 primary research interests are centered around 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 IFAC (Manfred Thoma Medal 2020) and EUCA (European Control Award 2020). His students were winners or finalists for Best Student Paper awards at the European Control Conference (2013, 2019), the American Control Conference (2016), the Conference on Decision and Control (2020), the PES General Meeting (2020), and the PES PowerTech Conference (2017). He is furthermore a recipient of the 2010 ACC 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 UCSB ME Best Ph.D. award.

REFERENCES

- [1] L. Ljung, "Identification for control: simple process models," in *Proc. 41st IEEE Conf. on Decision and Control*, 2002, vol. 4, Dec. 2002, pp. 4652–4657.
- [2] M. Gevers, "Identification for control: From the early achievements to the revival of experiment design," *European Journal of Control*, vol. 11, no. 4, pp. 335–352, 2005.
- [3] H. Hjalmarsson, "From experiment design to closed-loop control," *Automatica*, vol. 41, no. 3, pp. 393–438, 2005.
- [4] A. Feldbaum, "Dual control theory problems," *IFAC Proceedings Volumes*, vol. 1, no. 2, pp. 541–550, 1963.
- [5] M. Ferizbegovic, J. Umenberger, H. Hjalmarsson, and T. B. Schön, "Learning robust lq-controllers using application oriented exploration," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 19–24, 2019.
- [6] 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," *European Journal of Control*, vol. 29, pp. 1–16, 2016.
- [7] A. Iannelli, M. Khosravi, and R. S. Smith, "Structured exploration in the finite horizon linear quadratic dual control problem," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 959–964, 2020.
- [8] A. Chiuso and G. Pillonetto, "System identification: A machine learning perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 281–304, 2019.
- [9] S. Formentin and A. Chiuso, "Control-oriented regularization for linear system identification," *Automatica*, vol. 127, p. 109539, 2021.
- [10] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct & indirect data-driven control formulations via regularizations and relaxations," *IEEE Trans. Automat. Contr.*, 2023.
- [11] P. Stoica and Y. Selén, "Model-order selection: A review of information criterion rules," *IEEE Signal Proc. Magazine*, vol. 21, pp. 36–47, 2004.
- [12] J. C. Willems, "From time series to linear system—Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling," *Automatica*, vol. 22, 23, pp. 561–580, 675–694, 87–115, 1986, 1987.
- [13] J. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory*. Springer-Verlag, 1998.
- [14] J. C. Willems, "The behavioral approach to open and interconnected systems: Modeling by tearing, zooming, and linking," *Control Systems Magazine*, vol. 27, pp. 46–99, 2007.
- [15] —, "Terminals and ports," *IEEE Circuits and Systems Magazine*, vol. 10, no. 4, pp. 8–26, 2010.
- [16] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annual Reviews in Control*, vol. 52, pp. 42–64, 2021.

- [17] 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, 2014.
- [18] A. Damen, P. Van den Hof, and A. Hajdasinski, "Approximate realization based upon an alternative to the Hankel matrix: the Page matrix," *Control Sys. Lett.*, vol. 2, pp. 202–208, 1982.
- [19] S. Brunton, J. Proctor, and N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Academy of Sci.*, vol. 113, pp. 3932–3937, 2016.
- [20] J. C. Willems, P. Rapisarda, I. Markovsky, and B. De Moor, "A note on persistency of excitation," *Control Lett.*, vol. 54, no. 4, pp. 325–329, 2005.
- [21] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory, implementation, applications*. Boston: Kluwer, 1996.
- [22] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *Int. J. Control*, vol. 81, no. 12, pp. 1946–1959, 2008.
- [23] 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, 2005.
- [24] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, vol. 67, pp. 3289–3304, 2022.
- [25] V. Mishra, I. Markovsky, and B. Grossmann, "Data-driven tests for controllability," *Control Systems Letters*, vol. 5, pp. 517–522, 2020.
- [26] J. Berberich, J. Köhler, M. A. Muller, and F. Allgower, "Linear tracking mpc for nonlinear systems part ii: The data-driven case," *IEEE Transactions on Automatic Control*, 2022, To appear. Available at <https://ieeexplore.ieee.org/abstract/document/9756053>.
- [27] B. Nortmann and T. Mylvaganam, "Direct data-driven control of linear time-varying systems," arXiv:2111.02342, 2021.
- [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, 1966.
- [29] I. Markovsky, *Low-Rank Approximation: Algorithms, Implementation, Applications*, 2nd ed. Springer, 2019.
- [30] I. Markovsky and F. Dörfler, "Data-driven dynamic interpolation and approximation," *Automatica*, vol. 135, p. 110008, 2022.
- [31] —, "Identifiability in the behavioral setting," Tech. Rep., 2019.
- [32] T. Söderström, "Errors-in-variables methods in system identification," *Automatica*, vol. 43, pp. 939–958, 2007.
- [33] 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.
- [34] —, "Software for weighted structured low-rank approximation," *J. Comput. Appl. Math.*, vol. 256, pp. 278–292, 2014.
- [35] I. Markovsky, "A software package for system identification in the behavioral setting," *Control Eng. Practice*, vol. 21, pp. 1422–1436, 2013.
- [36] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Stanford University, 2002.
- [37] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel, "DAISY: A database for identification of systems," *Journal A*, vol. 38, no. 3, pp. 4–5, 1997, available from <http://homes.esat.kuleuven.be/~smc/daisy/>.
- [38] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd edition. Dover, 2008.
- [39] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annual Reviews in Control*, vol. 52, pp. 42–64, 2021.
- [40] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2019.
- [41] H. J. van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: a new perspective on data-driven analysis and control," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4753–4768, 2020.
- [42] J. Berberich, C. W. Scherer, and F. Allgower, "Combining prior knowledge and data for robust controller design," 2020, arXiv preprint arXiv:2009.05253.
- [43] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.
- [44] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [45] J. C. Willems, "In control, almost from the beginning until the day after tomorrow," *European Journal of Control*, vol. 13, no. 1, p. 71, 2007.
- [46] B. D. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. Courier Corporation, 2007.
- [47] J. Köhler, K. P. Wabersich, J. Berberich, and M. N. Zeilinger, "State space models vs. multi-step predictors in predictive control: Are state space models complicating safe data-driven designs?" *arXiv preprint arXiv:2203.15471*, 2022.
- [48] W. Favoreel, B. De Moor, and M. Gevers, "SPC: subspace predictive control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004–4009, 1999.
- [49] B. Huang and R. Kadali, *Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach*. Springer, 2008.
- [50] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1077–1091, 2005.
- [51] 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 *2012 American Control Conference*. IEEE, 2012, pp. 4239–4244.
- [52] E. Elokda, J. Coulson, P. N. Beuchat, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for quadcopters," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8916–8936, 2021.
- [53] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for grid-connected power converters," in *IEEE Conf. on Decision and Control*, 2019.
- [54] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct & indirect data-driven control formulations via regularizations and relaxations," January 2021, <https://arxiv.org/abs/2101.01273>.
- [55] F. Fiedler and S. Lucia, "On the relationship between data-enabled predictive control and subspace predictive control," in *European Control Conference*, 2021.
- [56] V. Breschi, A. Chiuso, and S. Formentin, "The role of regularization in data-driven predictive control," *arXiv preprint arXiv:2203.10846*, 2022.
- [57] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control – a computer control algorithm," in *joint automatic control conference*, no. 17, 1980, p. 72.
- [58] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice – A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [59] L. Hewing, K. Wabersich, M. Menner, and M. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.
- [60] A. Mesbah, K. P. Wabersich, A. P. Schoellig, M. N. Zeilinger, S. Lucia, T. A. Badgwell, and J. A. Paulson, "Fusion of machine learning and mpc under uncertainty: What advances are on the horizon?" in *American Control Conference (ACC 2022)*, 2022, pp. 342–357.
- [61] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.
- [62] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *European Control Conf.*, 2019, pp. 307–312.
- [63] F. Dörfler, P. Tesi, and C. De Persis, "On the certainty-equivalence approach to direct data-driven LQR design," *arXiv preprint arXiv:2109.06643*, 2021.
- [64] L. Huang, Z. Jianzhe, J. Lygeros, and F. Dörfler, "Robust data-enabled predictive control: Tractable formulations and performance guarantees," 2021, arXiv preprint arXiv:2105.07199.
- [65] V. Krishnan and F. Pasqualetti, "On direct vs indirect data-driven predictive control," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 736–741.
- [66] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Quadratic regularization of data-enabled predictive control: Theory and application to power converter experiments," in *IFAC Symposium on System Identification*, vol. 54, no. 7, 2021, pp. 192–197.
- [67] J. Berberich, J. Köhler, M. A. Müller, and F. Allgower, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2021.
- [68] A. Xue and N. Matni, "Data-driven system level synthesis," in *Learning for Dynamics and Control*. 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 Transactions on Automatic Control*, 2022, To appear. Available at <https://ieeexplore.ieee.org/abstract/document/9789718>.
- [70] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, "System level synthesis," *Annual Reviews in Control*, vol. 47, pp. 364–393, 2019.

- [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 Systems Letters*, vol. 3, no. 4, pp. 1014–1019, 2019.
- [72] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Decentralized data-enabled predictive control for power system oscillation damping," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 1065–1077, 2021.
- [73] M. Yin, A. Iannelli, and R. S. Smith, "Maximum likelihood estimation in data-driven modeling and control," *IEEE Transactions on Automatic Control*, 2021, To appear. Available at <https://ieeexplore.ieee.org/abstract/document/9661376>.
- [74] J. Coulson, J. Lygeros, and F. Dörfler, "Regularized and distributionally robust data-enabled predictive control," in *Proc. of IEEE Conf. on Decision and Control*, 2019, pp. 7165–7170.
- [75] 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, 2020.
- [76] —, "On the design of terminal ingredients for data-driven mpc," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 257–263, 2021.
- [77] J. Bongard, J. Berberich, J. Köhler, and F. Allgöwer, "Robust stability analysis of a simple data-driven model predictive control approach," *IEEE Transactions on Automatic Control*, 2022, To appear. Available at <https://ieeexplore.ieee.org/abstract/document/9744574>.
- [78] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Robust constraint satisfaction in data-driven MPC," in *2020 59th IEEE Conference on Decision and Control*, 2020, pp. 1260–1267.
- [79] D. Alpago, F. Dörfler, and J. Lygeros, "An extended kalman filter for data-enabled predictive control," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 994–999, 2020.
- [80] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.
- [81] R. Ou, G. Pan, and T. Faulwasser, "Data-driven multiple shooting for stochastic optimal control," *IEEE Control Systems Letters*, 2022.
- [82] F. Milano, F. Dörfler, G. Hug, D. J. Hill, and G. Verbič, "Foundations and challenges of low-inertia systems," in *2018 power systems computation conference (PSCC)*. IEEE, 2018, pp. 1–25.
- [83] B. Kroposki, B. Johnson, Y. Zhang, V. Gevorgian, P. Denholm, B.-M. Hodge, and B. Hannegan, "Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy," *IEEE Power and energy magazine*, vol. 15, no. 2, pp. 61–73, 2017.
- [84] F. Blaabjerg, Z. Chen, and S. B. Kjaer, "Power electronics as efficient interface in dispersed power generation systems," *IEEE transactions on power electronics*, vol. 19, no. 5, pp. 1184–1194, 2004.
- [85] M. Vali, V. Petrović, L. Y. Pao, and M. Kühn, "Model predictive active power control for optimal structural load equalization in waked wind farms," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 30–44, 2021.
- [86] L. Huang, H. Xin, Z. Li, P. Ju, H. Yuan, Z. Lan, and Z. Wang, "Grid-synchronization stability analysis and loop shaping for PLL-based power converters with different reactive power control," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 501–516, 2020.
- [87] H. Liu, X. Xie, J. He, T. Xu, Z. Yu, C. Wang, and C. Zhang, "Subsynchronous interaction between direct-drive pmsg based wind farms and weak ac networks," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4708–4720, 2017.
- [88] 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 transactions on Power Systems*, vol. 34, no. 2, pp. 1393–1403, 2018.
- [89] S. Golestan and J. M. Guerrero, "Conventional synchronous reference frame phase-locked loop is an adaptive complex filter," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1679–1682, 2014.
- [90] 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 Journal of Industry Applications*, vol. 1, pp. 115–134, 2020.
- [91] L. Grüne and A. Rantzer, "On the infinite horizon performance of receding horizon controllers," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2100–2111, 2008.
- [92] L. Huang, H. Xin, Z. Wang, K. Wu, H. Wang, J. Hu, and C. Lu, "A virtual synchronous control for voltage-source converters utilizing dynamics of dc-link capacitor to realize self-synchronization," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 5, no. 4, pp. 1565–1577, 2017.
- [93] 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.
- [94] C. Yang, L. Huang, H. Xin, and P. Ju, "Placing grid-forming converters to enhance small signal stability of PLL-integrated power systems," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3563–3573, 2020.
- [95] 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, 2013.
- [96] B. B. Johnson, S. V. Dhople, A. O. Hamadeh, and P. T. Krein, "Synchronization of parallel single-phase inverters with virtual oscillator control," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 6124–6138, 2013.
- [97] 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 Transactions on Automatic Control*, vol. 64, no. 11, pp. 4496–4511, 2019.
- [98] Y. Gu and T. C. Green, "Power system stability with a high penetration of inverter-based resources," *Proceedings of the IEEE*, 2022.
- [99] H. Gao, H. Xin, L. Huang, Z. Li, W. Huang, P. Ju, and C. Wu, "Common-mode frequency in inverter-penetrated power systems: Definition, analysis, and quantitative evaluation," *IEEE Transactions on Power Systems*, 2022.
- [100] P.-K. Keung, P. Li, H. Banakar, and B. T. Ooi, "Kinetic energy of wind-turbine generators for system frequency support," *IEEE Transactions on power systems*, vol. 24, no. 1, pp. 279–287, 2008.
- [101] L. Huang, H. Xin, L. Zhang, Z. Wang, K. Wu, and H. Wang, "Synchronization and frequency regulation of dfbg-based wind turbine generators with synchronized control," *IEEE Transactions on Energy Conversion*, vol. 32, no. 3, pp. 1251–1262, 2017.
- [102] 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," *arXiv preprint arXiv:2108.00153*, 2021.