Annual Reviews in Control 52 (2021) 42–64

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

Behavioral systems theory in data-driven analysis, signal processing, and control

Ivan Markovsky [a,*], Florian Dörfler [b]

[a] Department ELEC, Vrije Universiteit Brussel, Brussels, 1050, Belgium
[b] Automatic Control Laboratory (IfA), ETH Zürich, Zürich, 8092, Switzerland

ARTICLE INFO

Keywords:
Behavioral systems theory
Data-driven control
Missing data estimation
System identification

ABSTRACT

The behavioral approach to systems theory, put forward 40 years ago by Jan C. Willems, takes a representation-free perspective of a dynamical system as a set of trajectories. Till recently, it was an unorthodox niche of research but has gained renewed interest for the newly emerged data-driven paradigm, for which it is uniquely suited due to the representation-free perspective paired with recently developed computational methods. A result derived in the behavioral setting that became known as the fundamental lemma started a new class of subspace-type data-driven methods. The fundamental lemma gives conditions for a non-parametric representation of a linear time-invariant system by the image of a Hankel matrix constructed from raw time series data. This paper reviews the fundamental lemma, its generalizations, and related data-driven analysis, signal processing, and control methods. A prototypical signal processing problem, reviewed in the paper, is missing data estimation. It includes simulation, state estimation, and output tracking control as special cases. The direct data-driven control methods using the fundamental lemma and the non-parametric representation are loosely classified as implicit and explicit approaches. Representative examples are data-enabled predictive control (an implicit method) and data-driven linear quadratic regulation (an explicit method). These methods are equally amenable to certainty-equivalence as well as to robust control. Emphasis is put on the robustness of the methods under noise. The methods allow for theoretical certification, they are computationally tractable, in comparison with machine learning methods require small amount of data, and are robustly implementable in real-time on complex physical systems.

1. Introduction

The behavioral approach to system theory was put forward by Jan C. Willems in the early 1980s to resolve "many awkward things with input/output thinking" (Willems, 2007b, Section 8). In addition to enforcing "input/output thinking", conventional system theory approaches invariably associate a dynamical system with one of its representations, *e.g.*, a convolution, transfer function, or state–space representation. The new perspective brought by the behavioral approach separates the system from its numerous representations by defining a system as a set of trajectories. This abstract set-theoretic perspective makes the "input/output thinking" a choice rather than a requirement.

In addition to making the input/output thinking optional, separation of the system from its representations has other far reaching consequences. It gives a geometric view of a linear time-invariant system as a (low-dimensional) shift-invariant subspace in a (high-dimensional) trajectory space. This geometric viewpoint is often simpler and more natural than the classical frequency-domain and state–space ones. It led to a "clear and rational foundation under the problem of obtaining models from time series" (Willems, 1986, 1987). In particular, the global total least squares (Roorda & Heij, 1995), deterministic subspace (Van Overschee & De Moor, 1996), and structured low-rank approximation (Markovsky, 2013, 2019) approaches to system identification are motivated by Willems (1986, 1987). More recently, the behavioral approach contributed key ideas and techniques for data-driven analysis, signal processing, and control. This paper reviews these ideas and techniques, presents some of the methods that originate from them, and outlines research directions for future work.

In the contemporary language of machine learning, the new techniques are unsupervised and non-parametric. The techniques are unsupervised in the sense that they use directly the raw time-series data without labeling or pre-processing inputs and outputs, which require human decision making. The methods are non-parametric in the sense that they do not involve a parametric model representation

[*] Corresponding author.
E-mail addresses: imarkovs@vub.be (I. Markovsky), dorfler@ethz.ch (F. Dörfler).

https://doi.org/10.1016/j.arcontrol.2021.09.005
Received 15 July 2021; Received in revised form 26 September 2021; Accepted 27 September 2021
Available online 10 November 2021

of the data-generating system nor of the solution, *e.g.*, the filter or the controller.

The results reviewed in the paper originate from Markovsky (2017), Willems et al. (2005). The key result of Willems et al. (2005), which became known as the *fundamental lemma*, gives conditions for existence of a *non-parametric representation* of a discrete-time linear time-invariant system that is specified by a trajectory of the system. The second cornerstone for this paper is the idea put forward in Markovsky (2017) that various signal processing and control problems can be posed and solved as a single generic missing data estimation problem.

Recently, an ever-growing number of generalizations and applications of the fundamental lemma has appeared in the context of data-driven analysis and control. We provide a concise and comprehensive (albeit not exhaustive) review of the literature. A peculiar aspect of the literature centered around the fundamental lemma (and behavioral system theory in general) is that from the publication of the result in 2005 till 2018 the result remained unnoticed—as evident by the citation record—before it has blossomed in the wake of data-driven control methods.

In what follows, we provide a brief historical recap and contextualization of Willems et al. (2005). The fundamental lemma was originally conceived as a purely theoretical system identification result: it gives *identifiability conditions* for deterministic linear time-invariant systems. These conditions can be viewed also as *input design* guidelines. Under certain specified conditions on the input signal and the data-generating system, the collected data reveal the system dynamics. Informally, when the data is assembled in a Hankel matrix, it spans the set of all finite-length trajectories of the system, *i.e.*, it represents the finite-time behavior of the system. The identifiability conditions given by the fundamental lemma are important, however, in retrospect what led to the renewed interest is the non-parametric representation that gave rise to a new computational approach for solving data-driven analysis and design problems.

The non-parametric representation was used originally as a tool to find alternative derivations of existing subspace identification methods (Markovsky et al., 2005). It revealed the system theoretic meaning of computational steps in the methods. For example, the oblique projection was shown to compute sequential free responses of the system. Later on the non-parametric representation was used for solving other data-driven problems.

As a byproduct of the fundamental lemma, new algorithms for system identification, data-driven simulation, and data-driven control were proposed by Markovsky and Rapisarda (2008), Markovsky et al. (2006). The algorithms for data-driven simulation and control of Markovsky and Rapisarda (2008), which are a follow-up of the ones in Markovsky et al. (2005), are noteworthy because of:

- *simplicity*—they require only solving a system of linear equations that involves no hyperparameters,
- *generality*—apply to multivariable systems under arbitrary initial condition, and
- *robustness*—subsequent results show that with relatively minor modifications the algorithms "work" also for noisy data and nonlinear data-generating systems.

These features make the proposed data-driven simulation and control algorithms a viable and practical alternative to the conventional model-based approach that requires parametric model identification followed by model-based simulation and control.

Earlier precursors of the data-driven control methods originating from the fundamental lemma are the work of Favoreel et al. (1999) on *subspace predictive control* and the work of Fujisaki et al. (2004) and Ikeda et al. (2001), who proposed a conceptual framework for subspace-type data-driven control. In Section 5.1.2 we show the connection of the subspace predictive control to methods derived from the fundamental lemma. The framework of Fujisaki et al. (2004) and Ikeda et al. (2001) is also motivated by the behavioral perspective of a system as a set of trajectories and uses a non-parametric representation of the system's behavior similar to the ones shown in Section 3.3.

Subsequent work (Coulson et al., 2019a) built on the data-driven control algorithms of Markovsky and Rapisarda (2008) and embedded them in a robustified version of receding horizon predictive control. The resulting procedure, called *Data-EnablEd Predictive Control (DeePC),* saw many extensions and successful implementations in different application areas. Since 2019, Berberich and Allgöwer (2020), De Persis and Tesi (2019) and van Waarde et al. (2020) and others used the core result of the fundamental lemma for solving various analysis and control design problems directly from input-state data. The foundations laid by the fundamental lemma as well as the numerous adoptions and implementations led to a blossoming literature in *direct data-driven control*, which is reviewed in this paper.

The remainder of this paper is organized as follows. Section 2 gives a self-contained introduction to the behavioral approach, which is sufficient to derive in Section 3 methods for data-driven representation of linear time-invariant systems. These methods are applied in Section 4 for the solution of a generic missing data estimation problem and in Section 5 for solving direct data-driven control problems via predictive control or explicit feedback policies. Section 6 gives conclusions and poses open research directions. Selected, self-contained, and educational proofs are given in the appendices.

2. Behavioral system theory

The idea of separating the notion of a dynamical system from its representations is one of the hallmarks of the behavioral approach to system theory. It also plays an important role in data-driven system theory, signal processing, and control. Section 2.1 defines a dynamical system and in particular a linear time-invariant system as a set of trajectories. The class of linear time-invariant systems is refined in Section 2.2 by defining the notion of model complexity and the subclass of bounded complexity linear time-invariant systems. Three representations of bounded complexity linear time-invariant systems—kernel, input/output, and input/state/output—are presented in Section 2.3. A representation-free notion of controllability is introduced and linked to the classical notion of state controllability in Section 2.4. The simulation problem and specification of initial condition in a representation-free manner are presented in Section 2.5. Our presentation is self-contained and focused on what is needed for the coming sections on the fundamental lemma and data-driven control. For a more in-depth introduction to the behavioral approach, we refer to Polderman and Willems (1998) and Willems (1986, 1987, 1991, 2007a).

2.1. Dynamical systems as sets of trajectories

The concept of a dynamical system is fundamental in system theory. System theory textbooks, however, rarely define it rigorously. In the special case of a linear system, the concept of a dynamical system is linked to the one of an input–output map, defined by convolution or a transfer function if the system is also time-invariant. More advanced textbooks identify a dynamical system with the ubiquitous state–space equations.

The behavioral approach to system theory starts with a profound, yet simple idea: *a dynamical system is a set of trajectories.* This set is referred to as the *behavior*. Properties of the system as well as problems involving the system are defined in terms of the behavior. For example, a system is *linear* if it is a subspace and *time-invariant* if it is invariant to the action of the shift operator. Thus, a linear time-invariant system is a shift-invariant subspace. There is no a priori reference to inputs, outputs, and initial condition. However, the definition has the input–output map as a special case and responses due to nonzero initial condition are included in the behavior.

The behavior is all that matters: two systems are identical if and only if their behaviors are equal. How the system is specified is an

Table 1
Summary of notation.

Notation:	Definition:
$\mathbb{N} := \{\, 1, 2, \dots \,\}$	Set of natural numbers
$w \in (\mathbb{R}^q)^{\mathbb{N}},\ w : \mathbb{N} \to \mathbb{R}^q$	q-variate real discrete-time signal with time axis $\mathbb{N}$
$w\|_L := \big(w(1), \dots, w(L)\big)$	Restriction of w to the interval $[1, L]$
$w = w_{\mathrm{p}} \wedge w_{\mathrm{f}}$	Concatenation of trajectories
$\sigma,\ (\sigma w)(t) := w(t+1)$	Unit shift operator
$\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{N}}$	Discrete-time dynamical system $\mathscr{B}$ with q variables
$\mathscr{B}\|_L := \{\, w\|_L \mid w \in \mathscr{B} \,\}$	Restriction of $\mathscr{B}$ to the interval $[1, L]$
$\mathscr{L}^q$	Set of linear time-invariant systems with q variables
$\mathbf{m}(\mathscr{B})/\mathbf{l}(\mathscr{B})/\mathbf{n}(\mathscr{B})$	Number of inputs/lag/order of $\mathscr{B}$
$\mathbf{c}(\mathscr{B}) := \big(\mathbf{m}(\mathscr{B}), \mathbf{l}(\mathscr{B}), \mathbf{n}(\mathscr{B})\big)$	Complexity of $\mathscr{B}$
$\mathscr{L}^q_c := \{\, \mathscr{B} \in \mathscr{L}^q \mid \mathbf{c}(\mathscr{B}) \le c \,\}$	Set of bounded complexity Linear time-invariant systems
$\mathscr{H}_L(w)$	Hankel matrix with L block rows constructed from w, see (9)
$A^{\dagger}$	The pseudo-inverse of A

important but separate question. A specification of the system is called a *representation*. Thus, in the behavioral setting the concept of representation is decoupled from the one of a system. A system admits different representations. For example, convolution, transfer function, and state–space equations are representations of a linear time-invariant system and not the system itself. The system is the solution set of these equations. The view of the representations as incidental is a major departure point of the behavioral from the classical approach.

A valid criticism of the behavioral approach is that it is abstract and difficult to work with in practice. Relevant questions being asked are: What is the value of the set-theoretic formalism? What can be done with it that cannot be done in the classical setting by working with representations of the system? How can problems be solved in practice without using representations? Indeed, in practice representations are needed for specifying a system and for solving problems both analytically and computationally. The value of the set-theoretic formalism is on the higher level of defining properties and problems in a representation-free manner.

A representation-free problem formulation is important because it decouples the meaning and objectives of the problem from its potential solution methods. The choice of the representation should pertain to the solution method only. In addition, to the problem statement–solution method decoupling, the behavioral approach led to new representations, analysis tools, and design methods that have no counterparts in the input–output setting. Till recently, using the behavioral approach had primarily conceptual and theoretical benefits, as evident by a few practical algorithms that came out from it. This has changed in the last 15 years, when algorithms, software, and applications inspired by or developed for the behavioral approach emerged. The change was catalyzed by research on data-driven methods, which are both using and contributing to the representation-free perspective of dynamical systems.

Notation: The notation used in the rest of the paper is summarized in Table 1. The space of q-variate one-side infinite time-series (discrete-time signals) is denoted by $(\mathbb{R}^q)^{\mathbb{N}}$. Recall that a linear time-invariant (LTI) system $\mathscr{B}$ with q variables is a shift-invariant subspace of $(\mathbb{R}^q)^{\mathbb{N}}$. The set of q-variate linear time-invariant systems is denoted by $\mathscr{L}^q$. In the next section, we elaborate on the structure of a linear time-invariant system, introducing the notion of system's complexity and defining subsets of $\mathscr{L}^q$ of systems with bounded complexity.

2.2. Bounded complexity linear time-invariant systems

Apart from being a shift-invariant subspace, a linear time-invariant system $\mathscr{B}$ has additional structure described by a set of integers, called *integer invariants* or *structure indices* (Willems, 1986, Section 7). The most important ones are the *number of inputs (free variables)* $\mathbf{m}(\mathscr{B})$, the *lag* $\mathbf{l}(\mathscr{B})$, and the *order* $\mathbf{n}(\mathscr{B})$. These structure indices are inherent properties of the system, *i.e.*, they are independent of its representation. However, as shown in Section 2.3, the structure indices can be expressed also in terms of the parameters of an input/output, minimal kernel, and minimal state–space representations, where they are linked to familiar concepts.

The *restriction* $\mathscr{B}|_L$ of the system $\mathscr{B} \in \mathscr{L}^q$ to the interval $[1, L]$ is the set of L-samples long trajectories of $\mathscr{B}$. By linearity of $\mathscr{B}$, $\mathscr{B}|_L$ is a subspace of $\mathbb{R}^{qL}$. Its dimension $\dim \mathscr{B}|_L$ is

$$\dim \mathscr{B}|_L = \mathbf{m}(\mathscr{B})L + \mathbf{n}(\mathscr{B}), \quad \text{for } L \ge \mathbf{l}(\mathscr{B}). \tag{1}$$

The dimension formula (1) is used often in the rest of the paper. An easily accessible proof based on the familiar input/state/output representation of $\mathscr{B}$ is given in Appendix A. (We refer to Markovsky and Dörfler (2020) for a state–space independent proof.) Intuitively, $\dim \mathscr{B}|_L$ is the degrees of freedom in choosing a trajectory $w \in \mathscr{B}|_L$ of the system. The term $\mathbf{m}(\mathscr{B})L$ in (1) corresponds to the degrees of freedom due to the choice of the input, and the term $\mathbf{n}(\mathscr{B})$ corresponds to the degrees of freedom due to the choice of the initial condition.

The notion of complexity of a dynamical system is related to the "size" of its behavior—the more complex the system is, the more trajectories it allows. In the case of linear time-invariant systems, (1) characterizes the "size" of $\mathscr{B}$. The system is called *bounded complexity* if $\dim \mathscr{B}|_L < qL$ for sufficiently large L. Bounded complexity implies that not all signals in $(\mathbb{R}^q)^{\mathbb{N}}$ are trajectories of the system or, equivalently, that not all variables are inputs, *i.e.*, $\mathbf{m}(\mathscr{B}) < q$, and the order $\mathbf{n}(\mathscr{B})$ is finite. Since

$$\mathbf{l}(\mathscr{B}) \le \mathbf{n}(\mathscr{B}) \le \big(q - \mathbf{m}(\mathscr{B})\big)\mathbf{l}(\mathscr{B}), \tag{2}$$

the lag $\mathbf{l}(\mathscr{B})$ of a bounded complexity system $\mathscr{B}$ is also finite. Formulae (1) and (2) can be derived using the minimal kernel and input/state/output representations of the system, introduced in the next section. For details, see Willems (1986, Section 7).

Formally, we define the triple

$$\mathbf{c}(\mathscr{B}) := \big(\mathbf{m}(\mathscr{B}), \mathbf{l}(\mathscr{B}), \mathbf{n}(\mathscr{B})\big)$$

as the *complexity* of $\mathscr{B}$. Then,

$$\mathscr{L}^q_{(m,\ell,n)} := \{\, \mathscr{B} \in \mathscr{L}^q \mid \mathbf{m}(\mathscr{B}) \le m,\ \mathbf{l}(\mathscr{B}) \le \ell,\ \ \mathbf{n}(\mathscr{B}) \le n \,\}$$

is the set of *bounded complexity linear time-invariant systems*.

The restricted behavior $\mathscr{B}|_{\mathbf{l}(\mathscr{B})+1}$ of a bounded complexity linear time-invariant system $\mathscr{B}$ uniquely specifies the system (Markovsky & Dörfler, 2020, Lemma 12). Hence, if a certain property can be certified for the set of trajectories $\mathscr{B}|_L$ of finite length $L > \mathbf{l}(\mathscr{B})$, then it holds for all trajectories $\mathscr{B}$. A system $\mathscr{B}$ without inputs, *i.e.*, $\mathbf{m}(\mathscr{B}) = 0$, is called *autonomous*. For a linear time-invariant autonomous system $\dim \mathscr{B}|_L = \mathbf{n}(\mathscr{B})$ if and only if $L \ge \mathbf{l}(\mathscr{B})$.

2.3. Parametric representations of bounded complexity linear time-invariant systems

A bounded complexity linear time-invariant system $\mathscr{B} \in \mathscr{L}^q_{(m,\ell,n)}$ admits different *parametric representations*. Next, we review (1) the kernel representation, which is a higher order difference equation involving the variables, *i.e.*, an auto-regressive time-series model, (2) the input/output representation, which is a kernel representation with additional structure—partitioning of the variables into inputs and outputs—and (3) the input/state/output representation, which in addition to the input/output partitioning of the variables introduces an auxiliary (state) variable, a first order difference equation of the state, and a static relation among the output, input, and state.

- A *kernel representation* of the system $\mathscr{B}$ is defined by a linear constant-coefficients difference equation

$$\mathscr{B} = \ker\ R(\sigma) := \big\{\, w \mid R(\sigma)w = 0 \,\big\}, \tag{3}$$

where σ is the *unit shift operator*, $(\sigma w)(t) := w(t+1)$ and the operator $R(\sigma)$ is defined by the polynomial matrix

$$\begin{aligned} R(z) &= R_0 + R_1 z + \cdots + R_\ell z^\ell \\ &= \begin{bmatrix} R^1(z) \\ \vdots \\ R^k(z) \end{bmatrix} = \begin{bmatrix} R^1_0 + R^1_1 z + \cdots + R^1_{\ell_1} z^{\ell_1} \\ \vdots \\ R^k_0 + R^k_1 z + \cdots + R^k_{\ell_k} z^{\ell_k} \end{bmatrix} \in \mathbb{R}^{k\times q}[z]. \end{aligned}$$

The kernel representation (3) is called *minimal* if the number of equations k is as small as possible over all kernel representations of $\mathscr{B}$ (Willems, 1991, Definition III). In a minimal kernel representation, $k = p := q - \mathbf{m}(\mathscr{B})$—the number of outputs of $\mathscr{B}$—and $\ell := \deg R := \max_i \ell_i$ is also minimized over all kernel representations of $\mathscr{B}$. The minimal degree ℓ of R is equal to the lag $\mathbf{l}(\mathscr{B})$ of the system. The minimal total degree $n := \sum_{i=1}^{k} \ell_i$ of R is equal to the order $\mathbf{n}(\mathscr{B})$ of the system.

- *Input/output representation:* For a permutation matrix $\Pi \in \mathbb{R}^{q\times q}$ and an integer $m \in [1,q]$ define via

$$\begin{bmatrix} u \\ y \end{bmatrix} := \Pi^{-1} w \tag{4}$$

a *partitioning* of the variables $w(t) \in \mathbb{R}^q$ into $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$, where $p = q - m$. Let Π_u be the projection of w on the variable u, *i.e.*, $\Pi_u w := u$. Acting on a set, Π_u projects all elements in the set. The partitioning (4) is an *input/output partitioning* of $\mathscr{B}$, *i.e.*, u is an *input* and y is an *output* of the system, if (1) $\Pi_u \mathscr{B} = (\mathbb{R}^m)^{\mathbb{N}}$, *i.e.*, u is a free variable, (2) the output y is not anticipating the input u (Willems, 1991, Definition VIII.4), and (3) the number of inputs $m := \dim u$ is maximal over all partitionings (4) of $\mathscr{B}$ that satisfy properties 1 and 2.

Let $\mathscr{B} = \ker R(\sigma)$ be a minimal kernel representation. The partitioning (4) is an input/output partitioning of $\mathscr{B}$ if and only if with $\begin{bmatrix} Q & -P \end{bmatrix} := R\Pi$, where the polynomial matrix $P \in \mathbb{R}^{p\times p}[z]$ is non-singular (Willems, 1991). The resulting *input/output representation* (also called *polynomial matrix descriptions* (Antsaklis & Michel, 1997, Chapter 7.5)) is

$$\mathscr{B}_{\text{i/o}}(P,Q,\Pi) = \left\{ \Pi \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right] \mid Q(\sigma)u = P(\sigma)y \right\}.$$

- The *input/state/output representation* is defined as

$$\begin{aligned} \mathscr{B} = \mathscr{B}_{\text{ss}}&(A,B,C,D,\Pi) \\ &:= \Big\{ \Pi \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right] \mid \text{there is } x \in (\mathbb{R}^n)^{\mathbb{N}}, \text{ such that} \\ &\qquad \sigma x = Ax + Bu,\ y = Cx + Du \Big\}, \end{aligned} \tag{5}$$

where $\Pi \in \mathbb{R}^{q\times q}$ is a permutation and $\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right] \in \mathbb{R}^{(n+p)\times(n+m)}$. The input/state/output representation (5) is called *minimal* if $n := \dim A$ is as small as possible over all input/state/output representations of $\mathscr{B}$ (Willems, 1991, Definition VII.5). In a minimal input/state/output representation, the state dimension n is equal to the order $\mathbf{n}(\mathscr{B})$ of the system. The lag $\mathbf{l}(\mathscr{B})$ also manifests itself in a minimal input/state/output representations of $\mathscr{B}$. It is equal to the smallest k, for which the *extended observability matrix*

$$\mathscr{O}_k(A,C) := \operatorname{col}(C, CA, \dots, CA^{k-1}) \tag{6}$$

reaches full column rank, *i.e.*, $\operatorname{rank} \mathscr{O}_k(A,C) = n$.

In a minimal input/state/output representation (5), the pair (A,C) is *state observable*, *i.e.*, $\operatorname{rank} \mathscr{O}_{\mathbf{l}(\mathscr{B})}(A,C) = \mathbf{n}(\mathscr{B})$, however, the pair (A,B) need not be *state controllable*, *i.e.*,

$$\operatorname{rank} \begin{bmatrix} B & AB & \cdots & A^{\mathbf{l}(\mathscr{B})-1}B \end{bmatrix} = \mathbf{n}(\mathscr{B}). \tag{7}$$

As shown in the next section, (A,B) is state controllable if and only if the system $\mathscr{B}$ is controllable in a new representation-invariant sense of controllability.

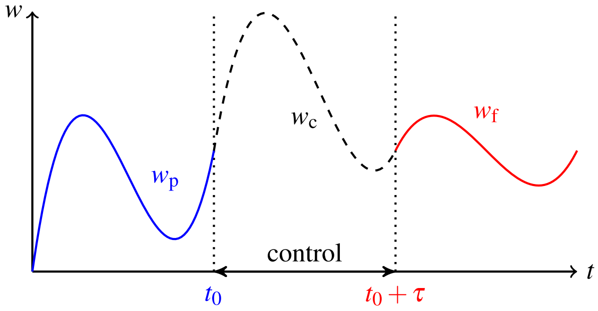

Fig. 1. A system is controllable if any "past" trajectory w_p can be concatenated with any "future" trajectory w_f via a suitable "control" trajectory w_c.

2.4. Controllability in the behavioral setting

As all system properties, in the behavioral setting, controllability is defined in terms of the behavior: $\mathscr{B}$ is *controllable* if for any $w_\text{p}, w_\text{f} \in \mathscr{B}$ and $t_0 \in \mathbb{N}$, there is $\tau \in \mathbb{N}$ and $w \in \mathscr{B}$, such that $w(t) = w_\text{p}(t)$, for $t \in [1, t_0]$, and $w(t) = w_\text{f}(t)$, for $t \geq t_0 + \tau$ (Willems, 1991, Definition V.1). *I.e.*, $\mathscr{B}$ is controllable if it is possible to "patch" any "past" trajectory w_p to any "future" trajectory w_f by including a control trajectory w_c over a period of length τ; see Fig. 1 for a visual illustration.

Note that the definition of controllability in the behavioral setting is not restricted to linear time-invariant systems: it applies to general dynamical systems. In Pillai and Willems (1999), it is used also in the context of multidimensional systems. When specialized to linear time-invariant systems, the notion of controllability in the behavioral setting is related but not equivalent to the classical notion of *state controllability* (7).

The state controllability is defined as a property of the pair of parameters (A,B) in an input/state/output representation $\mathscr{B}_\text{ss}(A,B,C,D)$ of a linear time-invariant system $\mathscr{B}$. It depends therefore on the choice of the representation (albeit it is invariant under similarity transformations) as well as on the properties of the system $\mathscr{B}$. A particular pair (A,B) may be uncontrollable because the input has no sufficient effect on the output or due to a "bad" choice of the state. Disentangling the two causes is an important benefit of using the behavioral setting. Controllability in the behavioral setting is a property of a system $\mathscr{B}$ only. Thus, it is independent of the choice of the representation.

In order to show the relation between the behavioral and the classical notions of controllability, consider a linear time-invariant system $\mathscr{B} \in \mathscr{L}^q$ with an input/state/output representation $\mathscr{B} = \mathscr{B}_\text{ss}(A,B,C,D)$. Without loss of generality we assume that the parameters (A,B,C,D) are in the Kalman decomposition form:

$$A = \begin{bmatrix} A_{c\bar{o}} & \star & \star & \star \\ 0 & A_{co} & 0 & \star \\ 0 & 0 & A_{\bar{c}\bar{o}} & \star \\ 0 & 0 & 0 & A_{\bar{c}o} \end{bmatrix}, \quad B = \begin{bmatrix} B_{c\bar{o}} \\ B_{co} \\ 0 \\ 0 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & C_{co} & 0 & C_{\bar{c}o} \end{bmatrix}.$$

The system $\mathscr{B}$ has a direct sum decomposition $\mathscr{B} = \mathscr{B}_\text{ctr} \oplus \mathscr{B}_\text{aut}$ into a controllable subsystem $\mathscr{B}_\text{ctr} \in \mathscr{L}^q$ and an autonomous subsystem $\mathscr{B}_\text{aut} \in \mathscr{L}^q$, see Willems (1991, Proposition V.8). The controllable subsystem $\mathscr{B}_\text{ctr}$ corresponds to the subsystem $\mathscr{B}_\text{ss}(A_{co}, B_{co}, C_{co}, D)$ in the Kalman decomposition, and the autonomous subsystem $\mathscr{B}_\text{aut}$ corresponds to $\mathscr{B}_\text{ss}(A_{\bar{c}o}, 0, C_{\bar{c}o}, 0)$. Therefore, in order to preserve the behavior, unobservable states in a state–space representation can be removed, but uncontrollable states that are observable should not be removed. Indeed, the unobservable states have no contribution to $\mathscr{B}$, however, the uncontrollable states that are observable define the autonomous subsystem $\mathscr{B}_\text{aut}$ and removing them changes $\mathscr{B}$.

The fact that uncontrollable–observable states cannot be removed clashes with the classical wisdom that "minimality" of a state–space

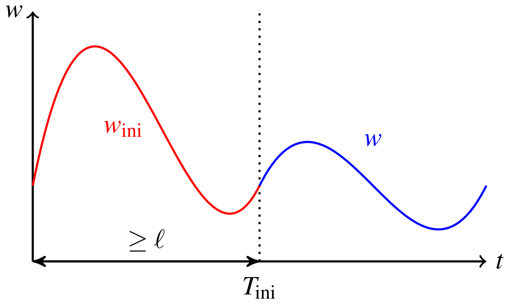

Fig. 2. Initial condition for a trajectory $w \in \mathscr{B}$ are specified in the behavioral setting by a prefix trajectory w_{ini} of length $T_{\text{ini}} \geq \mathbf{l}(\mathscr{B})$. The condition that w is generated from the initial condition specified by w_{ini} is then $w_{\text{ini}} \wedge w \in \mathscr{B}$.

representation of a system implies both state observability and state controllability. A quadruple (A, B, C, D) that is both state observable and state controllable is called a *minimal realization* of the system. The surprising fact brought by the behavioral approach is that some nonminimal realizations are in fact real; equivalently, there are "real-life" systems that do not admit minimal state–space realization. The behavioral notion of controllability clarifies the folklore behind the "without loss of generality" assumption of existence of a minimal realization as well as the cancellation of common poles and zeros in a transfer function. In both questions—when a minimal realization exists and when a pole-zero cancellation is allowed—the answer is "when the system is controllable".

2.5. Simulation and specification of initial condition

Simulation of a dynamical system is one of the basic operations in the arsenal of system theory. It is defined for a system with an input/output partitioning of the variables as follows: Finding the output of the system, given the system, the input, and the initial condition. From a mathematical point of view, simulation is the problem of solving an equation, *e.g.*, (3), (2.3), or (5) if the system $\mathscr{B}$ is given by on of the representations reviewed in Section 2.3. From the behavioral point of view, simulation is particular way to parametrize a trajectory $w \in \mathscr{B}$ of the system $\mathscr{B}$, *i.e.*, abstractly viewed, simulation is the problem of selecting an element of the behavior. A convenient way of achieving parametrization of $w \in \mathscr{B}$ is to fix an input/output partitioning (4) of the variables, and take as parameters the input component u of $w = \Pi \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$ and the initial condition. As shown next, the initial condition can be specified also by a trajectory—a "prefix" trajectory w_{ini} for w.

Formally, the simulation problem is defined as follows: Given a system $\mathscr{B}$, an input/output partitioning (4), input $u \in (\mathbb{R}^m)^L$, and initial condition $w_{\text{ini}} \in (\mathbb{R}^q)^{T_{\text{ini}}}$,

$$\text{find } y \in (\mathbb{R}^p)^L, \text{ such that } w_{\text{ini}} \wedge \Pi(u, y) \in \mathscr{B}|_{T_{\text{ini}}+L}, \tag{8}$$

see Fig. 2 for a graphical illustration.

The assumption that w_{ini} is a trajectory of $\mathscr{B}$ guarantees existence of a solution to the simulation problem (8). However, in general, the solution may not be unique. In order to render the solution unique, w_{ini} has to be "sufficiently" long. A sufficient condition for this is that $T_{\text{ini}} \geq \mathbf{l}(\mathscr{B})$.

Lemma 1 (*Initial Condition Specification* Markovsky & Rapisarda, 2008**).** *Let $\mathscr{B} \in \mathscr{L}^q$ admit an input/output partition $w = (u, y)$. Then, for any given $w_{ini} \in \mathscr{B}_{T_{ini}}$ with $T_{ini} \geq \mathbf{l}(\mathscr{B})$ and $u \in (\mathbb{R}^m)^L$, there is a unique $y \in (\mathbb{R}^p)^L$, such that $w_{ini} \wedge (u, y) \in \mathscr{B}|_{T_{ini}+L}$.*

Proof. Two proofs of Lemma 1 are given in Appendix B. The first one is based on an input/state/output representation of $\mathscr{B}$. The second one is based on a generic basis of $\mathscr{B}|_{T_{\text{ini}}+L}$. □

The trajectory w_{ini} plays the role of the initial state in the state–space setting. It can be shown that for $T_{\text{ini}} \geq \mathbf{l}(\mathscr{B})$, the vector w_{ini} of sequential samples is a state vector of the system (Rapisarda & Willems, 1997). Then, problems of estimation of w_{ini} can also be understood as state estimation problems.

3. Data-driven non-parametric model representation

This section presents a non-parametric representation of linear time-invariant systems that is at the core of the subspace-type data-driven analysis, estimation, and control methods. The non-parametric representation is the image of a Hankel matrix constructed from trajectories of the system. It has been used implicitly since the 90's in subspace identification and related data matrices are extensively used in dictionary learning and motion primitives. A theoretical foundation for its use, however, was provided only later by the so-called *fundamental lemma*. Apart from providing a foundation for the non-parametric representation, the fundamental lemma, reviewed in Section 3.1, gives also identifiability conditions, *i.e.*, conditions under which the data-generating system can be uniquely identified from the data. While the fundamental lemma provides sufficient conditions for identifiability and for the non-parametric representation from an input design perspective, Section 3.2 presents alternative necessary and sufficient conditions based on the rank of the Hankel matrix of the data. Section 3.3 previews two ways of using the non-parametric representation for solving data-driven analysis, signal processing, and control problems.

3.1. The fundamental lemma

Given one "long" trajectory $w_{\text{d}} \in (\mathbb{R}^q)^T$ of a linear time-invariant system $\mathscr{B} \in \mathscr{L}^q$, multiple "short" trajectories of $\mathscr{B}$ can be created exploiting time-invariance. In what follows, the subscript "d" stands for "data" and indicates one or more trajectories of a system that are used to implicitly specify it. Let $L \in \{1, \ldots, T\}$ be the length of the "short" trajectories and define the cut operator

$$w|_L := \big(w(1), \ldots, w(L)\big).$$

Sequential application of the shift σ and cut $|_L$ operators on w_{d} results in $N = T - L + 1$, L-samples-long trajectories

$$(\sigma^0 w_{\text{d}})|_L, (\sigma^1 w_{\text{d}})|_L, \ldots, (\sigma^{T-L} w_{\text{d}})|_L.$$

The $(qL) \times (T - L + 1)$-dimensional matrix

$$\mathscr{H}_L(w_{\text{d}}) := \begin{bmatrix} w_{\text{d}}(1) & w_{\text{d}}(2) & \cdots & w_{\text{d}}(T-L+1) \\ w_{\text{d}}(2) & w_{\text{d}}(3) & \cdots & w_{\text{d}}(T-L+2) \\ \vdots & \vdots & & \vdots \\ w_{\text{d}}(L) & w_{\text{d}}(L+1) & \cdots & w_{\text{d}}(T) \end{bmatrix}, \tag{9}$$

formed by stacking these trajectories next to each other is called the *Hankel matrix* of w_{d} (with depth L). Although $\mathscr{H}_L(w_{\text{d}})$ is well defined for any $L \in \{1, \ldots, T\}$, we require $\mathscr{H}_L(w_{\text{d}})$ to have more columns than rows, which implies

$$L \leq L_{\max} := \left\lfloor \tfrac{T+1}{q+1} \right\rfloor,$$

where $\lfloor a \rfloor$ is the largest integer smaller than or equal to a.

A signal $u_{\text{d}} \in (\mathbb{R}^m)^T$ is called *persistently exciting of order* L if $\mathscr{H}_L(u_{\text{d}})$ is full row rank, *i.e.*, $\operatorname{rank} \mathscr{H}_L(u_{\text{d}}) = mL$. It follows from (1) that a persistently exciting signal u_{d} of order L cannot be modeled as a trajectory of an autonomous linear time-invariant system with lag less than L. For u_{d} to be persistently exciting of order L, it must be sufficiently rich and long. In particular, it must have at least $T_{\min} := (m+1)L - 1$ samples. Persistency of excitation plays an important role in system identification and input design problems.

The following result, which became known as the fundamental lemma, gives both identifiability conditions as well as input (experiment) design guidelines.

Lemma 2 (*Fundamental Lemma* Willems et al., 2005). *Consider a linear time-invariant system $\mathscr{B} \in \mathscr{L}^q$ with an input/output partition $w = (u, y)$. Let*

1. *$w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d}) \in \mathscr{B}|_T$ be a trajectory of $\mathscr{B}$,*
2. *the system $\mathscr{B}$ be controllable, and*
3. *the input component u_d of w_d be persistently exciting of order $L + \mathbf{n}(\mathscr{B})$.*

Then, any L-samples long trajectory w of $\mathscr{B}$ can be written as a linear combination of the columns of $\mathscr{H}_L(w_\mathrm{d})$ and any linear combination $\mathscr{H}_L(w_\mathrm{d})g$, for $g \in \mathbb{R}^{T-L+1}$, is also a trajectory of $\mathscr{B}$, i.e.,

$$\operatorname{image} \mathscr{H}_L(w_\mathrm{d}) = \mathscr{B}|_L. \tag{10}$$

About the proof: While the inclusion $\operatorname{image} \mathscr{H}_L(w_\mathrm{d}) \subseteq \mathscr{B}|_L$ follows directly from the linearity and time-invariance properties of the system, the question when equality holds is not obvious. The original proof in Willems et al. (2005) is based on a kernel representation of the system and uses the notion of annihilators of the behavior. The tool used in this proof is abstract algebra. An alternative proof based on a state–space representation is given in van Waarde, De Persis et al. (2020). Both proofs are by contradiction. Quoting from Willems et al. (2005):

> "The interesting, and somewhat surprising, part of Theorem 1 is that persistency of excitation of order $L + \mathbf{n}(\mathscr{B})$ is needed in order to be able to deduce that the observed sequences 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 $(u_\mathrm{d}, y_\mathrm{d})$ which are considered".

Currently there is no constructive proof that gives an intuition why the additional persistency of excitation is needed nor how conservative the conditions are.

Lemma 2 states conditions on the input u_d and the system $\mathscr{B}$ under which, independently of the initial condition corresponding to w_d, the Hankel matrix $\mathscr{H}_L(w_\mathrm{d})$ spans the restricted behavior $\mathscr{B}|_L$. Therefore, the image of the Hankel matrix $\mathscr{H}_L(w_\mathrm{d})$ is a representation of the system $\mathscr{B}$, as long as trajectories of length L are concerned. The resulting representation (10) is non-parametric, applies to controllable linear time-invariant systems, and depends on the given data w_d only. Although (10) is a representation of the restricted behavior $\mathscr{B}|_L$, the fact that it involves raw data only and no parameters distinguishes it from conventional parametric and non-parametric system representations that are traditionally called "models". In this sense, direct data-driven methods using only (10) are "model-free".

Over time, Lemma 2 became known as the *fundamental lemma* because of its foundational importance for system identification, data-driven analysis, signal processing, and control. Indeed, since $\mathscr{B}|_{\mathbf{l}(\mathscr{B})+1}$ completely specifies the system $\mathscr{B}$, choosing $L = \mathbf{l}(\mathscr{B}) + 1$ in Lemma 2, we obtain conditions under which $\mathscr{B}$ can be recovered from the data w_d, *i.e.*, identifiability conditions. Moreover, the result is interpretable and as shown later on in the overview, algorithms derived from it are tractable and robust.

A special case of the fundamental lemma that derives identifiability condition used in subspace identification (Van Overschee & De Moor, 1996; Verhaegen & Dewilde, 1992) is when the system $\mathscr{B}$ is given by an input/state representation, *i.e.*, the output is equal to the state:

$$\mathscr{B} = \{\, (u, x) \mid \sigma x = Ax + Bu \,\}. \tag{11}$$

Note that in this case $\mathbf{l}(\mathscr{B}) = 1$. Applied to (11), Lemma 2 leads to the following result (Willems et al., 2005, Corollary 2).

Corollary 3. Consider a trajectory $(u_\mathrm{d}, x_\mathrm{d}) \in \mathscr{B}|_T$ of a system (11) with $m := \dim u$ inputs and state dimension $n := \dim x$. Assume that (A, B) is controllable. Then,

1. if u_d is persistently exciting of order n,
$$\operatorname{rank} \begin{bmatrix} x_\mathrm{d}(1) & \cdots & x_\mathrm{d}(T) \end{bmatrix} = n,$$
2. if u_d is persistently exciting of order $n + 1$,
$$\operatorname{rank} \begin{bmatrix} u_\mathrm{d}(1) & \cdots & u_\mathrm{d}(T) \\ x_\mathrm{d}(1) & \cdots & x_\mathrm{d}(T) \end{bmatrix} = n + m,$$
3. if u_d is persistently exciting of order $n + L$,
$$\operatorname{rank} \begin{bmatrix} \mathscr{H}_L(u_\mathrm{d}) \\ \begin{bmatrix} x_\mathrm{d}(1) & \cdots & x_\mathrm{d}(T-L+1) \end{bmatrix} \end{bmatrix} = mL + n.$$

The rank conditions appearing in Corollary 3 are extensively used in the subspace identification literature, however, they are assumed and there were no tests available to verify them from given input–output data $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d})$ that is an arbitrary trajectory. Corollary 3 gives such a test. Corollary 3 is also extensively used in data-driven analysis and control, see Section 5.3.

The fundamental lemma has been generalized for uncontrollable systems (Mishra et al., 2020; Yu et al., 2021), data consisting of multiple trajectories (van Waarde, De Persis et al., 2020), other matrix structures (Coulson et al., 2020), as well as the following model classes: affine (Berberich et al., 2021c), linear parameter-varying (Verhoek et al., 2021), flat systems (Alsalti et al., 2021), finite impulse response Volterra (Rueda-Escobedo & Schiffer, 2020), and Wiener–Hammerstein (Berberich & Allgöwer, 2020). Other works extending the fundamental lemma to nonlinear systems use the Koopman operator (Lian & Jones, 2021a; Lian, Wang et al., 2021). Experiment design methods for the fundamental lemma, *i.e.*, methods for choosing the trajectory w_d, are considered in De Persis and Tesi (2021a), Iannelli et al. (2020) and van Waarde (2021).

Like the fundamental lemma, however, all above cited generalizations depend on a priori given input/output partitioning of the variables and provide sufficient conditions in terms of persistency of excitation of the input. The following section presents an alternative result that relaxes the assumption of a given input/output partitioning. It expresses the persistency of excitation in terms of all variables, provides necessary and sufficient conditions not assuming controllability, and widens the class of data matrix structures.

3.2. Identifiability

More generally, instead of one trajectory w_d, consider a set

$$\mathscr{W}_\mathrm{d} := \{\, w_\mathrm{d}^1, \ldots, w_\mathrm{d}^N \,\}, \quad w_\mathrm{d}^i \in (\mathbb{R}^q)^{T_i} \tag{12}$$

of N trajectories of a dynamical system $\mathscr{B}$, *i.e.*,

$$w_\mathrm{d}^i \in \mathscr{B}|_{T_i}, \quad \text{for all } i = 1, \ldots, N. \tag{13}$$

We refer to $\mathscr{W}_\mathrm{d}$ as the *data* and to $\mathscr{B}$ as the *data-generating system*. The question "Can we recover the data-generating system $\mathscr{B}$ from the data $\mathscr{W}_\mathrm{d}$?" is called the *identifiability* question. In order to make the identifiability question well posed, it is necessary to know in addition to the data $\mathscr{W}_\mathrm{d}$ a model class $\mathscr{M}$ to which the to-be-identified system $\mathscr{B}$ belongs. In this paper this is the linear time-invariant model class $\mathscr{L}^q$ or $\mathscr{L}_c^q$ that uses prior knowledge of an upper bound c on the complexity.

Definition 4 (*Identifiability*). The system $\mathscr{B} \in \mathscr{M}$ is identifiable from the data (12)–(13) in the model class $\mathscr{M}$ if $\mathscr{B}$ is the only model in $\mathscr{M}$ that fits the data exactly, *i.e.*,

$$\widehat{\mathscr{B}} \in \mathscr{M} \text{ and } \mathscr{W}_\mathrm{d} \subset \widehat{\mathscr{B}} \quad \Longrightarrow \quad \widehat{\mathscr{B}} = \mathscr{B}.$$

Identifiability gives conditions for well-posedness of the *exact identification problem*, which is the map $\mathscr{W}_\mathrm{d} \mapsto \widehat{\mathscr{B}} \in \mathscr{M}$ from data to a model in the model class.

For a set of time series $\mathscr{W}_\mathrm{d}$, the Hankel matrix (9) is generalized to the *mosaic-Hankel matrix* (Heinig, 1995; Usevich & Markovsky, 2014)

$$\mathscr{H}_L(\mathscr{W}_\mathrm{d}) := \begin{bmatrix} \mathscr{H}_L(w_\mathrm{d}^1) & \cdots & \mathscr{H}_L(w_\mathrm{d}^N) \end{bmatrix}.$$

In Markovsky and Dörfler (2020), the following identifiability condition for a linear time-invariant data-generating system $\mathscr{B} \in \mathscr{L}^q$ is proven: the system $\mathscr{B}$ is identifiable from the data (12)–(13) if and only if

$$\operatorname{rank} \mathscr{H}_{\mathbf{l}(\mathscr{B})+1}(\mathscr{W}_\mathrm{d}) = \mathbf{m}(\mathscr{B})\big(\mathbf{l}(\mathscr{B})+1\big) + \mathbf{n}(\mathscr{B}).$$

The following corollary of the identifiability condition provides a foundation for a non-parametric representation of the restricted behavior $\mathscr{B}|_L$ of the data-generating system.

Corollary 5 (*Corollary 19, Markovsky & Dörfler, 2020*). If the data-generating system $\mathscr{B}$ is linear time-invariant,

$$\operatorname{image} \mathscr{H}_L(\mathscr{W}_\mathrm{d}) \subseteq \mathscr{B}|_L, \quad \text{for all } L \in \{\, \mathbf{l}(\mathscr{B})+1, \dots, L_\mathrm{max} \,\}.$$

Moreover, for $L \geq \mathbf{l}(\mathscr{B})$, $\operatorname{image} \mathscr{H}_L(\mathscr{W}_\mathrm{d}) = \mathscr{B}|_L$ if and only if

$$\operatorname{rank} \mathscr{H}_L(\mathscr{W}_\mathrm{d}) = \mathbf{m}(\mathscr{B})L + \mathbf{n}(\mathscr{B}). \tag{14}$$

Proof. A representation-free proof is given in Appendix C. The key argument is showing that the dimension of the image of the Hankel matrix $\mathscr{H}_L(\mathscr{W}_\mathrm{d})$ is equal to the dimension of $\mathscr{B}|_L$. Then, (14) follows from the dimension formula (1). □

Corollary 5 is an alternative to the fundamental lemma. Like the fundamental lemma, it gives conditions under which the image of the Hankel matrix $\mathscr{H}_L(\mathscr{W}_\mathrm{d})$ constructed from the data generates the restricted behavior $\mathscr{B}|_L$. Unlike the fundamental lemma, however, Corollary 5 does not require a given input/output partitioning of the variables nor controllability of the data-generating system. Also, Corollary 5 gives a necessary and sufficient condition while the fundamental lemma gives sufficient conditions only. Condition (14) is reminiscent to the persistency of excitation condition in the fundamental lemma. We refer to it as a *generalized persistency of excitation*. It is verifiable from data $\mathscr{W}_\mathrm{d}$ and prior knowledge of the structure indices $\mathbf{m}(\mathscr{B})$, $\mathbf{l}(\mathscr{B})$, and $\mathbf{n}(\mathscr{B})$. An experiment design problem achieving the condition (14) with a minimal length input is addressed by van Waarde (2021).

Another way in which Corollary 5 generalizes the fundamental lemma is that it allows for multiple trajectories (12) as in van Waarde, De Persis et al. (2020). The mosaic-Hankel matrix in Corollary 5 includes as special cases other matrix structures, such as the Hankel matrix (9), the Page matrix, and the trajectory matrix. The *trajectory matrix* used in dictionary learning (Brunton et al., 2016) collects time series column-by-column as

$$\mathscr{T}_L(\mathscr{W}_\mathrm{d}) := \begin{bmatrix} w_\mathrm{d}^1(1) & w_\mathrm{d}^2(1) & \cdots & w_\mathrm{d}^N(1) \\ \vdots & \vdots & & \vdots \\ w_\mathrm{d}^1(L) & w_\mathrm{d}^2(L) & \cdots & w_\mathrm{d}^N(L) \end{bmatrix} \in \mathbb{R}^{qL\times N} \tag{15}$$

and is a special case of the mosaic-Hankel matrix when all time series w_d^i have length $T_1 = \cdots = T_N = L$, *i.e.*, $w_\mathrm{d}^i \in (\mathbb{R}^q)^L$ for all $i \in \{\, 1, \dots, N \,\}$. Coined by Damen et al. (1982), the *Page matrix* $\mathscr{P}_L(w_\mathrm{d}) \in \mathbb{R}^{qL\times T'}$ of the signal $w_\mathrm{d} \in (\mathbb{R}^q)^T$ with L block rows is a special trajectory matrix (and therefore a special mosaic-Hankel matrix) obtained by taking $w_\mathrm{d}^i = (\sigma^{(i-1)L} w_\mathrm{d})|_L$, for $i \in \{\, 1, \dots, T' \,\}$, where $T' := \lfloor T/L \rfloor$. Alternatively, the Page matrix $\mathscr{P}_L(w_\mathrm{d})$ can be obtained from the Hankel matrix $\mathscr{H}_L(w)$ by column selection:

$$\begin{aligned} \mathscr{P}_L(w_\mathrm{d}) &:= \begin{bmatrix} w_\mathrm{d}|_L & (\sigma^L w_\mathrm{d})|_L & \cdots & (\sigma^{(T'-1)L} w_\mathrm{d})|_L \end{bmatrix} \\ &= \begin{bmatrix} w_\mathrm{d}(1) & w_\mathrm{d}(L+1) & \cdots & w_\mathrm{d}\big((T'-1)L+1\big) \\ \vdots & \vdots & & \vdots \\ w_\mathrm{d}(L) & w_\mathrm{d}(2L) & \cdots & w_\mathrm{d}(T'L) \end{bmatrix}. \end{aligned} \tag{16}$$

Like the Hankel matrix $\mathscr{H}_L(w_\mathrm{d})$, the Page matrix $\mathscr{P}_L(w_\mathrm{d})$ also consists of L-samples long trajectories, however, unlike the Hankel matrix, the Page matrix has no repeated elements on the anti-diagonals. The Page matrix has been independently derived as a basis for the system behavior in Agarwal et al. (2018) and Coulson et al. (2020).

Some pros and cons of the different matrix structures are as follows: the Hankel matrix conditions the data on time-invariance leading to larger dimensional data matrices, whereas the trajectory and Page matrices offer algorithmic advantages since they are unstructured. Clearly the latter require more data. Markovsky and Dörfler (2020) report empirical results showing advantages of the Hankel matrix when used for system identification, while in Section 5 we show that the trajectory and Page matrices have advantages when used in direct data-driven control. We will touch upon these points in later sections and continue focusing on the mosaic-Hankel matrix $\mathscr{H}_L(\mathscr{W}_\mathrm{d})$ keeping the special matrix structures in mind. The essential property of all data matrices is that every column of the matrix, viewed as a time series, is an L-samples long trajectory of the system.

3.3. Data-driven representation of the restricted behavior

The fundamental lemma and Corollary 5 provide a non-parametric representation of the restricted behavior $\mathscr{B}|_L$ as the image of the mosaic-Hankel matrix $\mathscr{H}_L(\mathscr{W}_\mathrm{d})$ of the data with depth L. Under the generalized persistency of excitation condition (14) (or conditions 1–3 of Lemma 2),

$$\mathscr{B}|_L = \operatorname{image} \mathscr{H}_L(\mathscr{W}_\mathrm{d}). \tag{17}$$

For given L, the non-parametric data-driven representation (17) is completely specified by the data $\mathscr{W}_\mathrm{d}$. Under (14), it is valid for any multivariable linear time-invariant system. Note that the alternative conditions 1–3 of the fundamental lemma restrict the class of systems due to the controllability assumption.

Based on (17), two approaches for solving data-driven analysis, signal processing, and control problems were proposed:

1. solving a system of linear equations, and
2. solving a rank-constrained matrix approximation and completion problem.

The first approach, originally used for data-driven simulation and open-loop linear quadratic tracking control in Markovsky and Rapisarda (2008), expresses the constraint that $w \in (\mathbb{R}^q)^L$ is a trajectory of the system $\mathscr{B}$ as existence of a solution of a system of linear equation:

$$w \in \mathscr{B}|_L \quad \Longleftrightarrow \quad w = \mathscr{H}_L(\mathscr{W}_\mathrm{d})g \text{ has a solution } g. \tag{18}$$

The right-hand-side condition of (18) involves only the collected data, so that the system $\mathscr{B}$ need not be known. The approach using (18) requires basic linear algebra—the solution of a system of linear equations. For details see Section 4.3, where (18) is used for data-driven missing data estimation. Modifications of the approach for noisy data $\mathscr{W}_\mathrm{d}$ are presented in Section 4.4 and in Section 5 for data-driven control.

The second approach expresses the constraint that $w \in (\mathbb{R}^q)^L$ is a trajectory of a bounded complexity linear time-invariant system $\mathscr{B}$ as a rank condition:

$$w \in \mathscr{B}|_L \quad \Longleftrightarrow \quad \operatorname{rank} \begin{bmatrix} \mathscr{H}_\delta(\mathscr{W}_\mathrm{d}) \mathscr{H}_\delta(w) \end{bmatrix} = \operatorname{rank} \mathscr{H}_\delta(\mathscr{W}_\mathrm{d}), \quad \text{for any } \delta \in \{\, \mathbf{l}(\mathscr{B})+1, \dots, L \,\}. \tag{19}$$

The right-hand-side of (19) involves only the data and the parameter δ, so that again the system $\mathscr{B}$ need not be known. The condition is valid for any value of δ in the interval $[\mathbf{l}(\mathscr{B})+1, L]$, however, different choices of δ lead to different methods. Of most interest are the extremes $\delta = L$ and $\delta = \mathbf{l}(\mathscr{B})+1$. The case $\delta = L$ recovers the system of equations (17) approach, while the case $\delta = \mathbf{l}(\mathscr{B})+1$ leads to the mosaic-Hankel structured rank-constrained matrix approximation and completion approach of Markovsky (2008). In Section 4 we use (19) with $\delta = \mathbf{l}(\mathscr{B})+1$ for data-driven missing data estimation and in Section 5 the rank constraint (19) is used for pre-processing.

In (19), for $\delta = \mathbf{l}(\mathscr{B}) + 1$, the generalized persistency of excitation condition coincides with the identifiability condition. In contrast, in (18), L is the length of the signal w, which is given and is in general larger than $\mathbf{l}(\mathscr{B}) + 1$. The required generalized persistency of excitation for using the system of equations approach (18) is therefore more restrictive than the one for using the matrix completion approach (19) with $\delta = \mathbf{l}(\mathscr{B}) + 1$. Also, growing L implies growing dimension of the system of equations, which increases the computational cost.

Both deficiencies of (18)—the higher persistency of excitation and the higher computational cost—can be overcome by splitting w into length-δ pieces and computing each piece separately using (18), matching the initial condition of one piece with the final conditions of the previous piece, see Markovsky et al. (2005, Lemma 3). The resulting algorithm requires solving recursively a sequence of smaller dimensional systems of linear equations rather than one larger system. In the limit, the lengths of the pieces can be taken as $\delta = \mathbf{l}(\mathscr{B}) + 1$, which ensures that the required persistency of excitation is the same as the one for identifiability of the data-generating system $\mathscr{B}$. In this case, however, (18) is equivalent to a one-step-ahead predictor, which is essentially a model-based solution approach. The recursive algorithm outlined above is therefore a more flexible solution method that uses a δ-steps ahead data-driven prediction, where δ is a hyperparameter.

4. Data-driven missing data estimation

Apart from the data-driven representation that emerged from the fundamental lemma, the paper is based on another key idea, put forward in Markovsky (2017): a missing part of a generic trajectory of the system can be used to represent and compute the object that is aimed at, *e.g.*, the predicted signal in forecasting problems and the input signal in control problems. The seeds for this idea can be traced back to Markovsky and Rapisarda (2008). In Markovsky and Rapisarda (2008) two seemingly different problems—simulation and control—are solved by minor variations of the same basic method. It gradually emerged that this similarity is not incidental but a manifestation of a more general principle: rank deficiency of a structured data matrix. Consequently, the problem is structured matrix low-rank approximation and completion (Markovsky, 2014).

The problem considered in Markovsky (2017) deals with noisy data and its solution requires local optimization methods. In contrast, the problems in Markovsky and Rapisarda (2008) are for exact data and analytical solutions are derived. This disconnect between the two approaches is unfortunate. Latter research showed that the two approaches are complementary. In the context of data-driven control, the results in Markovsky (2017) were generalized for noisy data, using regularization methods. In Dörfler et al. (2021b) and Markovsky and Dörfler (2021) the low-rank approximation/completion approach and the regularization approaches are unified.

This section presents the missing data estimation problem. First, Section 4.1 shows how familiar model-based problems, such as simulation, Kalman smoothing, and output tracking control can be viewed as missing data estimation. Then, Section 4.2 presents a generic formulation of the missing data estimation problem that fits the examples as special cases. The generic problem is data-driven, *i.e.*, instead of a system a set of trajectories $\mathscr{W}_\text{d}$ is given. Section 4.3 outlines two solution approaches assuming that the data $\mathscr{W}_\text{d}$ is exact. The first solution is based on a rank constrained matrix approximation and completion reformulation of the problem that uses (19). The second solution is based on the system of linear equations reformulation of the problem that uses (18). Section 4.4 shows modifications of the methods for the case of inexact/noisy data $\mathscr{W}_\text{d}$. Numerical case studies are shown in Section 4.5. Finally, Section 4.6 comments on application of the data-driven methods for linear time-invariant system analysis.

4.1. Conventional model-based problem formulations

In order to fit conventional problems that are defined in terms of inputs and outputs in the behavioral setting, in this section we partition the variables w into inputs u and outputs y. For simplicity, we assume that $w = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$, *i.e.*, in (4) $\Pi = I$. Also, in order to specify or estimate initial condition in a representation-free manner, we split the time axis into "past" w_ini—the first T_ini samples—and "future" w_f—the remaining T_f samples (see Section 2.5). Then, by Lemma 1, we take $T_\text{ini} \geq \mathbf{l}(\mathscr{B})$.

Our goal is to show how conventional problems, such as simulation, smoothing, and output tracking control are equivalent to corresponding missing data estimation problems. For example, the simulation problem defined in (8) was already posed as missing data estimation: find the unknown y_f from the given initial condition w_ini and input u_f. The main message of this section is that the missing data estimation framework goes beyond simulation. Next, we show that it fits also two versions of state estimation—Kalman smoothing and errors-in-variables Kalman smoothing—as well as a output tracking control.

- *Kalman smoothing* The problem is defined as follows: given a linear time-invariant system $\mathscr{B}$ and a "noisy" trajectory w_f, find the initial condition w_ini. In the conventional Kalman smoothing problem the "noisy" trajectory w_f is generated in the *output error setup:* the output is measured with additive noise $y_\text{f} = \overline{y}_\text{f} + \widetilde{y}_\text{f}$, while the input is assumed exact $u_\text{f} = \overline{u}_\text{f}$. The true value $\overline{w}_\text{f}$ of the trajectory w_f is generated by the system $\mathscr{B}$ from some unknown true initial condition $\overline{w}_\text{ini}$, *i.e.*, $\overline{w}_\text{ini} \wedge \overline{w}_\text{f} \in \mathscr{B}|_{T_\text{ini}+T_\text{f}}$. Assuming further on that the measurement noise $\widetilde{y}_\text{f}$ is zero mean, white, Gaussian with covariance matrix that is a multiple of the identity, the maximum-likelihood estimation problem for the initial condition $\overline{w}_\text{ini}$ is given by
$$\text{minimize} \quad \text{over } \widehat{w}_\text{ini} \text{ and } \widehat{y}_\text{f} \quad \|y_\text{f} - \widehat{y}_\text{f}\|_2 \qquad \text{subject to} \quad \widehat{w}_\text{ini} \wedge (u_\text{f}, \widehat{y}_\text{f}) \in \mathscr{B}|_{T_\text{ini}+T_\text{f}}. \tag{20}$$
A byproduct of estimating the initial condition $\widehat{w}_\text{ini}$ in (20) is an approximation of the output $\widehat{y}_\text{f}$. The signal $\widehat{y}_\text{f}$ is the best estimate of the true output $\overline{y}_\text{f}$, given the model $\mathscr{B}$ and the prior knowledge about the measurement noise. Problem (20), which defines the conventional Kalman smoother (Kailath et al., 2000), is also a missing data estimation problem for $\overline{w}_\text{ini}$, however, the output y_f is approximated rather than fitted exactly.
- *Errors-in-variables (EIV) Kalman smoothing* The output error setup used in the conventional Kalman smoothing problem is asymmetric in the observed variables: the output is assumed noisy while the input is assumed exact. A symmetric setup where all variables are treated on an equal footing as noisy is called *errors-in-variables*. The errors-in-variables setup is consistent with the behavioral approach where all variables are treated on an equal footing without splitting them into inputs and outputs. The state estimation problem in the errors-in-variables setup is again: given a linear time-invariant system $\mathscr{B}$ and a "noisy" trajectory w_f, find the initial condition w_ini, however, now the "noisy" trajectory w_f is $w_\text{f} = \overline{w}_\text{f} + \widetilde{w}_\text{f}$, where $\overline{w}_\text{ini} \wedge \overline{w}_\text{f} \in \mathscr{B}|_{T_\text{ini}+T_\text{f}}$ for some $\overline{w}_\text{ini}$ and a zero mean, white, Gaussian noise $\widetilde{w}_\text{f}$ with covariance matrix that is a multiple of the identity. The maximum-likelihood estimation problem for the initial condition $\overline{w}_\text{ini}$ is then:
$$\text{minimize} \quad \text{over } \widehat{w}_\text{ini} \text{ and } \widehat{w}_\text{f} \quad \|w_\text{f} - \widehat{w}_\text{f}\|_2 \qquad \text{subject to} \quad \widehat{w}_\text{ini} \wedge \widehat{w}_\text{f} \in \mathscr{B}|_{T_\text{ini}+T_\text{f}}. \tag{21}$$
Problem (21) defines the what is called errors-in-variables Kalman smoother (Markovsky & De Moor, 2005). It is a missing data estimation problem for $\overline{w}_\text{ini}$, where the whole given trajectory w_f is approximated.

Table 2
The examples considered can be viewed as estimation of a missing part of a trajectory (the question marks "?"s in the table) where other parts of the trajectory are given as exact (E) or inexact/noisy (N).

Example	Reference	w_{ini}	u_{f}	y_{f}
Simulation	(8)	E	E	?
Kalman smoothing	(20)	?	E	N
EIV Kalman smoothing	(21)	?	N	N
Output tracking	(22)	E	?	N

Table 3
The information about exact, noisy, reference, and missing data elements $w_i(t)$ is encoded into the weights $v_i(t)$ of the element-wise weighted semi-norm $\|\cdot\|_v$.

Weight	Used if	To	By
$v_i(t) = \infty$	$w_i(t)$ exact	Interpolate $w_i(t)$	$e_i(t) = 0$
$v_i(t) \in (0, \infty)$	$w_i(t)$ noisy/reference	Approximate $w_i(t)$	$\min \|v_i(t)e_i(t)\|_2$
$v_i(t) = 0$	$w_i(t)$ missing	Fill in $w_i(t)$	$\hat{w} \in \widehat{\mathscr{B}}\|_L$

- *Output tracking* Finally, the least-squares output tracking problem is defined as follows: given initial condition w_{ini}, and an output y_{f}, find an input $\hat{u}_{\text{f}}$, such that

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{u}_{\text{f}} \text{ and } \hat{y}_{\text{f}} \quad \|y_{\text{f}} - \hat{y}_{\text{f}}\|_2 \\ &\text{subject to} \quad w_{\text{ini}} \wedge (\hat{u}_{\text{f}}, \hat{y}_{\text{f}}) \in \mathscr{B}|_{T_{\text{ini}}+T_{\text{f}}}. \end{aligned} \tag{22}$$

  The signal $\hat{u}_{\text{f}}$ is the *open-loop* optimal control signal. Problem (22) is a missing data estimation problem, where the missing data is the input. The given data is the reference signal y_{f}, which is approximated in the least-squares sense by the output $\hat{y}_{\text{f}}$. In the special case when the reference output $y_{\text{f}} \in \mathscr{B}|_{T_{\text{f}}}$, the problem is called *output-matching*. The output-matching problem is dual to the simulation problem, where the missing data is the output y_{f} and the given data is w_{ini} and the (exact) input u_{f}.
  In Section 5.1, we address the data-driven control extensions of (22). Note that the errors-in-variables Kalman smoothing problem (21) (with a more general weighted 2-norm cost function) is equivalent to the linear–quadratic tracking control problem (36), defined in Section 5.1.

Table 2 summarizes the examples. The data-driven versions of these signal processing problems assume given data $\mathscr{W}_{\text{d}}$ of the system $\mathscr{B}$. Consequently, the data-driven solution methods avoid identifying a parametric representation of $\mathscr{B}$.

4.2. Generic missing data problem formulation

In all problems considered in Section 4.1 the goal is to minimize the error signal $e := w - \hat{w}$, where w contains the given data (exact, noisy, or reference signal) as well as missing values and $\hat{w}$ is a trajectory of the system. The information about exact, noisy, reference, and missing data is encoded in the weights $v_i(t) \geq 0$ of the element-wise weighted semi-norm (see Table 3)

$$\|e\|_v := \sqrt{\sum_{t=1}^{L} \sum_{i=1}^{q} v_i(t) e_i^2(t)}.$$

Note that the given noisy and reference data is treated in the same way by approximating it in the weighted least squares sense. The difference is in the interpretation of the weights. In the noisy case and using the maximum-likelihood estimation principle, the weights are determined by the inverse of the noise variances, which are assumed a priori known. In the reference tracking case, the weights define the control objective, which is specified by the designer.

The examples considered are then special cases of the following generic missing data estimation problem

$$\text{minimize} \quad \text{over } \hat{w} \quad \|w - \hat{w}\|_v \quad \text{subject to} \quad \hat{w} \in \mathscr{B}|_L \tag{23}$$

for suitable choices of the trajectory w and the weights v. The formulation (23) not only generalizes the problems considered in Section 4.1 but can also be used to formulate other problems such as simulation with terminal conditions, trajectory generation with way points, and estimation of missing data (Markovsky & Dörfler, 2021).

In order to solve (23) numerically, we reformulate it as an equality constrained least-squares minimization. Let $\mathscr{I}_{\text{exact}}$ be the vector of indices of the exact given elements and $\mathscr{I}_{\text{tba}}$ be the vector of indices of the to-be-approximated (tba) given noisy or reference elements. We overload the notation $w|_L$ for a vector of indices $\mathscr{I} \in \{1, \ldots, qL\}^K$

$$w|_{\mathscr{I}} := \begin{bmatrix} w_{\mathscr{I}_1} & \cdots & w_{\mathscr{I}_K} \end{bmatrix}^\top \in \mathbb{R}^K$$

as the subvector of $w \in \mathbb{R}^{qL}$ with indices $\mathscr{I}$. Similarly, for latter usage, $\mathscr{H}_L(w_{\text{d}})|_{\mathscr{I}}$ is the submatrix of $\mathscr{H}_L(w_{\text{d}})$ with row indices $\mathscr{I}$. With this notation in place, the missing data estimation problem (23) becomes:

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{w} \quad \|w|_{\mathscr{I}_{\text{tba}}} - \hat{w}|_{\mathscr{I}_{\text{tba}}}\|_{v|_{\mathscr{I}_{\text{tba}}}} \\ &\text{subject to} \quad \hat{w} \in \mathscr{B}|_L \quad \text{and} \quad \hat{w}|_{\mathscr{I}_{\text{exact}}} = w|_{\mathscr{I}_{\text{exact}}}. \end{aligned} \tag{24}$$

In the next section we present methods for solving (24) based on the data-driven representation (17) of the system $\mathscr{B}$.

4.3. Solution methods with exact data $\mathscr{W}_{\text{d}}$

As previewed in Section 3.3, there are two distinct approaches: one by solving a system of linear equations and one by solving a rank-constrained matrix approximation and completion problem. In this section, we use them for solving the missing data estimation problem (24). We begin with the rank-constrained matrix approximation and completion approach.

By means of Corollary 5 and (19), we obtain a data-driven version of the missing data estimation problem (24):

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{w} \quad \|w|_{\mathscr{I}_{\text{tba}}} - \hat{w}|_{\mathscr{I}_{\text{tba}}}\|_{v|_{\mathscr{I}_{\text{tba}}}} \\ &\text{subject to} \quad \operatorname{rank} \begin{bmatrix} \mathscr{H}_\delta(\mathscr{W}_{\text{d}}) & \mathscr{H}_\delta(\hat{w}) \end{bmatrix} = \operatorname{rank} \mathscr{H}_\delta(\mathscr{W}_{\text{d}}) \\ &\qquad\qquad \text{and} \quad \hat{w}|_{\mathscr{I}_{\text{exact}}} = w|_{\mathscr{I}_{\text{exact}}}. \end{aligned} \tag{25}$$

I.e., assuming that (14) holds, the missing data estimation problem (24) is equivalent to the mosaic-Hankel structured low-rank matrix approximation and completion problem (25) for any $\delta \in \{\mathbf{l}(\mathscr{B}) + 1, \ldots, L\}$. The hyperparameter δ of (25) determines the shape of the Hankel matrix. In case of exact data, it does not affect the solution.

An independent yet similar data-driven approach to output tracking (22) was conceptually laid out by Ikeda et al. (2001). The authors also base their approach on the rank condition (14) to specify the rank of the Hankel matrix containing past data, the future output reference, and the future inputs to be designed.

Due to the rank constraint, (25) is a nonconvex optimization problem. A convex relaxation, based on the *nuclear norm* $\|\cdot\|_*$ regularization (see Fazel, 2002) is

$$\begin{aligned} &\text{minimize} \quad \text{over } \hat{w} \quad \|w|_{\mathscr{I}_{\text{tba}}} - \hat{w}|_{\mathscr{I}_{\text{tba}}}\|_{v|_{\mathscr{I}_{\text{tba}}}} \\ &\qquad\qquad + \gamma \left\| \begin{bmatrix} \mathscr{H}_\delta(\mathscr{W}_{\text{d}}) & \mathscr{H}_\delta(\hat{w}) \end{bmatrix} \right\|_* \\ &\text{subject to} \quad \hat{w}|_{\mathscr{I}_{\text{exact}}} = w|_{\mathscr{I}_{\text{exact}}}, \end{aligned} \tag{26}$$

where δ and γ are hyperparameters. The parameter γ controls the trade-off between the approximation error $\|w|_{\mathscr{I}_{\text{tba}}} - \hat{w}|_{\mathscr{I}_{\text{tba}}}\|_{v|_{\mathscr{I}_{\text{tba}}}}$ and the nuclear norm of the Hankel matrix, which is a surrogate for the system's complexity. Generally, γ should be chosen large enough in order to ensure the desired rank (19). In Dreesen and Markovsky (2019) it is suggested to consider a weighted data matrix $\begin{bmatrix} \alpha\mathscr{H}_\delta(\mathscr{W}_{\text{d}}) & \mathscr{H}_\delta(\hat{w}) \end{bmatrix}$, where $\alpha \geq 1$. It is shown that for α above certain threshold the solution of (26) coincides with the solution of (25), *i.e.*, the missing data is recovered exactly by solving (26).

The other approach for solving (24) is to use the linear equations representation (18). It leads to the equality-constraint least-squares problem (Markovsky & Dörfler, 2021)

$$\begin{aligned} &\text{minimize} \quad \text{over } g \text{ and } \widehat{w} \quad \| w|_{\mathscr{I}_{\text{tba}}} - \widehat{w}|_{\mathscr{I}_{\text{tba}}} \|_{v|_{\mathscr{I}_{\text{tba}}}} \\ &\text{subject to} \quad \widehat{w} = \mathscr{H}_L(\mathscr{W}_{\text{d}}) g \quad \text{and} \quad \widehat{w}|_{\mathscr{I}_{\text{exact}}} = w|_{\mathscr{I}_{\text{exact}}}, \end{aligned} \tag{27}$$

which admits a closed-form solution (Golub & Van Loan, 1996, Chapter 12). Note also that (27) has no hyperparameters. In case of uniform weights (*i.e.*, v_i = constant, for all $i \in \mathscr{I}_{\text{tba}}$) and no exact data (*i.e.*, $\mathscr{I}_{\text{exact}} = \emptyset$), the solution of (27) is

$$\widehat{w} = \mathscr{H}_L(\mathscr{W}_{\text{d}}) \big(\mathscr{H}_L(\mathscr{W}_{\text{d}})|_{\mathscr{I}_{\text{tba}}} \big)^{\dagger} w|_{\mathscr{I}_{\text{tba}}}, \tag{28}$$

where $M^{\dagger}$ is the pseudo-inverse of M.

The approach (27) using the linear equations representation is at first glance superior to the one using the low-rank Hankel matrix approximation and completion (25) due to the simplicity of the solution (28) and due its effective modifications for the case of inexact data presented in the next section. In case of inexact data, however, a modification of (25) also leads to methods for computing the statistically optimal maximum-likelihood estimator in the errors-in-variables setup (Markovsky, 2017).

4.4. Solution methods with inexact/noisy data $\mathscr{W}_d$

In Section 4.3, we assumed that $\mathscr{W}_{\text{d}}$ is exact. In this section, we assume that $\mathscr{W}_{\text{d}}$ as well as $w|_{\mathscr{I}_{\text{tba}}}$ are noisy and are generated in the *errors-in-variables* setup:

$$w_{\text{d}}^i = \overline{w}_{\text{d}}^i + \widetilde{w}_{\text{d}}^i, \text{ for } i = 1, \dots, N \quad \text{and} \quad w|_{\mathscr{I}_{\text{tba}}} = \overline{w}|_{\mathscr{I}_{\text{tba}}} + \widetilde{w}|_{\mathscr{I}_{\text{tba}}},$$

where $\overline{w}_{\text{d}}^i$, $\overline{w}|_{\mathscr{I}_{\text{tba}}}$ are the *true values* of w_{d}^i, $w|_{\mathscr{I}_{\text{tba}}}$, respectively, and $\widetilde{w}_{\text{d}}^i$, $\widetilde{w}|_{\mathscr{I}_{\text{tba}}}$ are the *measurement noises* that are assumed to be zero mean, Gaussian with joint covariance matrix $\big(\text{diag}(v_{\text{d}}^1, \dots, v_{\text{d}}^N, v|_{\mathscr{I}_{\text{tba}}})\big)^{-1}$. The true values of the signals are exact trajectories of a linear time-invariant system $\mathscr{B}$ with complexity bounded by $c = (m, \ell, n)$, *i.e.*,

$$\overline{w}_{\text{d}}^i \in \mathscr{B}|_{T_i}, \text{ for } i = 1, \dots, N, \quad \overline{w} \in \mathscr{B}_L, \quad \text{and} \quad \mathscr{B} \in \mathscr{L}_c^q. \tag{29}$$

The maximum-likelihood estimation problem in the errors-in-variables setup (29) is (Markovsky, 2017)

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{\mathscr{W}}_{\text{d}}, \widehat{w}, \text{ and } \widehat{\mathscr{B}} \quad \sum_{i=1}^{N} \| w_{\text{d}}^i - \widehat{w}_{\text{d}}^i \|_{v_{\text{d}}^i}^2 \\ &\qquad\qquad\qquad\qquad\qquad + \| w|_{\mathscr{I}_{\text{tba}}} - \widehat{w}|_{\mathscr{I}_{\text{tba}}} \|_{v|_{\mathscr{I}_{\text{tba}}}}^2 \\ &\text{subject to} \quad \widehat{w}_{\text{d}}^i \in \widehat{\mathscr{B}}|_{T_i} \quad \text{for } i = 1, \dots, N, \quad \widehat{w} \in \widehat{\mathscr{B}}|_L, \\ &\qquad\qquad \widehat{\mathscr{B}} \in \mathscr{L}_c, \quad \text{and} \quad \widehat{w}|_{\mathscr{I}_{\text{exact}}} = w|_{\mathscr{I}_{\text{exact}}}. \end{aligned} \tag{30}$$

By using the data-driven complexity characterization (19) and the fact that $\text{rank}\, \mathscr{H}_{\ell+1}(\overline{\mathscr{W}_{\text{d}}}) = (\ell + 1)m + n$, we restate the maximum-likelihood estimation problem (30) in a data-driven fashion as a mosaic-Hankel structured low-rank approximation and completion problem:

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{\mathscr{W}}_{\text{d}} \text{ and } \widehat{w} \quad \sum_{i=1}^{N} \| w_{\text{d}}^i - \widehat{w}_{\text{d}}^i \|_{v_{\text{d}}^i}^2 \\ &\qquad\qquad\qquad\qquad + \| w|_{\mathscr{I}_{\text{tba}}} - \widehat{w}|_{\mathscr{I}_{\text{tba}}} \|_{v|_{\mathscr{I}_{\text{tba}}}}^2 \\ &\text{subject to} \quad \text{rank} \begin{bmatrix} \mathscr{H}_{\ell+1}(\widehat{\mathscr{W}}_{\text{d}}) & \mathscr{H}_{\ell+1}(\widehat{w}) \end{bmatrix} \leq m(\ell+1) + n \\ &\qquad\qquad \text{and} \quad \widehat{w}|_{\mathscr{I}_{\text{exact}}} = w|_{\mathscr{I}_{\text{exact}}}. \end{aligned} \tag{31}$$

The latter is a nonconvex optimization problem due to the rank constraint. Local optimization methods based on the variable projections are developed in Markovsky and Usevich (2013). Suboptimal solution methods presented later on are:

1. a sequential two-step model-based approach and
2. convex relaxations based on sparse regularization.

The approach based on the linear equations representation (18) leads to the problem

$$\begin{aligned} &\text{minimize} \quad \text{over } g \quad \| w|_{\mathscr{I}_{\text{tba}}} - \mathscr{H}_L(\widehat{\mathscr{W}}_{\text{d}}^{\star})|_{\mathscr{I}_{\text{tba}}} g \|_{v|_{\mathscr{I}_{\text{tba}}}} \\ &\text{subject to} \quad \widehat{\mathscr{W}}_{\text{d}}^{\star} \in \arg\min_{\widehat{\mathscr{W}}_{\text{d}}} \sum_{i=1}^{N} \| w_{\text{d}}^i - \widehat{w}_{\text{d}}^i \|_{v_{\text{d}}^i}^2 \\ &\qquad\qquad\qquad \text{subject to} \quad \text{rank}\, \mathscr{H}_{\ell+1}(\widehat{\mathscr{W}}_{\text{d}}) \leq m(\ell+1) + n, \end{aligned} \tag{32}$$

which is also non-convex due to the rank constraint in the inner optimization. Problem (32) is a bi-level program: the inner level is estimation of $\overline{\mathscr{W}}_{\text{d}}$ and the outer level is estimation of the missing data $\overline{w}|_{\mathscr{I}_{\text{missing}}}$ using the estimate $\widehat{\mathscr{W}}_{\text{d}}^{\star}$ of $\overline{\mathscr{W}}_{\text{d}}$. Similar to (31), the bi-level problem (32) is amenable to either a sequential two-step procedure or a convex relaxation based on sparse regularization.

Two-step procedure: preprocessing of $\mathscr{W}_d$

The bi-level optimization problem (32) generally cannot be separated in two independent problems. Indeed, the solution of the outer problem depends on the inner problem and, in general, $\widehat{\mathscr{W}}_{\text{d}}^{\star}$ depends on the given data $w|_{\mathscr{I}_{\text{given}}}$, which includes the to-be-approximated $w|_{\mathscr{I}_{\text{tba}}}$ and the exact $w|_{\mathscr{I}_{\text{exact}}}$ samples, as well as on $\mathscr{W}_{\text{d}}$.

A heuristic two-step procedure estimates $\overline{\mathscr{W}}_{\text{d}}$ using $\mathscr{W}_{\text{d}}$ only:

1. preprocess $\mathscr{W}_{\text{d}}$, aiming to remove the noise, and
2. using the “cleaned” signal $\widehat{\mathscr{W}}_{\text{d}}$, find $w|_{\mathscr{I}_{\text{missing}}}$.

Thus, the two-step procedure reduces the problem with inexact data to the already solved problem with exact data.

The maximum-likelihood estimation of $\overline{\mathscr{W}}_{\text{d}}$ from $\mathscr{W}_{\text{d}}$ and the prior knowledge (29) is

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{\mathscr{W}}_{\text{d}} \text{ and } \widehat{\mathscr{B}} \quad \sum_{i=1}^{N} \| w_{\text{d}}^i - \widehat{w}_{\text{d}}^i \|_{v_{\text{d}}^i}^2 \\ &\text{subject to} \quad \widehat{w}_{\text{d}}^i \in \widehat{\mathscr{B}}|_{T_i} \text{ for } i = 1, \dots, N \quad \text{and} \quad \widehat{\mathscr{B}} \in \mathscr{L}_c^q. \end{aligned} \tag{33}$$

The formulation (33), however, is still a nonconvex optimization problem. For its solution, we use the SLRA package (Usevich & Markovsky, 2014), which is based on local optimization and computes as a byproduct an estimate $\widehat{\mathscr{B}}$ of the data generating system. Thus the two-step procedure becomes a model-based approach for solving (30): (1) using $\mathscr{W}_{\text{d}}$, identify a model $\widehat{\mathscr{B}}$, (2) using $\widehat{\mathscr{B}}$ and $w|_{\mathscr{I}_{\text{given}}}$, do model-based estimation of $w|_{\mathscr{I}_{\text{missing}}}$ (problem (24), using $\widehat{\mathscr{B}}$).

Although in general, the two-step procedure is suboptimal, when $\dim \mathscr{I}_{\text{given}} \leq mL + n$, the problem decouples and the two-step procedure is optimal, *i.e.*, the solution of (30) coincides with the solution of (33) followed by (24).

A suboptimal heuristic for preprocessing $\mathscr{W}_{\text{d}}$ is to perform unstructured low-rank approximation of the Hankel matrix $\mathscr{H}_L(\mathscr{W}_{\text{d}})$ by truncation of the singular value decomposition (SVD). The resulting Algorithm 1 does not derive a parametric model $\widehat{\mathscr{B}}$ of $\mathscr{B}$ and thus may be referred to as data-driven. Algorithm 1 requires prior knowledge of the number of inputs m and the order n, however, it has no other hyperparameters.

Algorithm 1 Data-driven missing data estimation with low-rank approximation preprocessing.

```
Input: $\mathscr{W}_d$, $\mathscr{I}_{given}$, $w|_{\mathscr{I}_{given}}$, $m$, and $n$.
  1: Compute the SVD: $\mathscr{H}_L(\mathscr{W}_d) = U \Sigma V^{\top}$.
  2: Let $r := mL + n$ and let $P \in \mathbb{R}^{qL \times r}$ be the submatrix of $U$ consisting
     of its first $r$ columns.
  3: Compute $\widehat{w} := P(P|_{\mathscr{I}_{given}})^{\top} w|_{\mathscr{I}_{given}}$.
Output: $\widehat{w}$.
```

Regularized least-squares approaches

Other approximation methods for missing data estimation with inexact data $\mathscr{W}_{\mathrm{d}}$ are the nuclear norm relaxation of (31)

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{\mathscr{W}}_{\mathrm{d}} \text{ and } \hat{w} \quad \sum_{i=1}^{N} \|w_{\mathrm{d}}^i - \hat{w}_{\mathrm{d}}^i\|_{v_{\mathrm{d}}^i}^2 \\ &\qquad + \|w|_{\mathscr{I}_{\mathrm{tba}}} - \hat{w}|_{\mathscr{I}_{\mathrm{tba}}}\|_{v|_{\mathscr{I}_{\mathrm{tba}}}}^2 + \gamma \left\| \begin{bmatrix} \mathscr{H}_{\delta}(\widehat{\mathscr{W}}_{\mathrm{d}}) & \mathscr{H}_{\delta}(\hat{w}) \end{bmatrix} \right\|_* \\ &\text{subject to} \quad \hat{w}|_{\mathscr{I}_{\mathrm{exact}}} = w|_{\mathscr{I}_{\mathrm{exact}}} \end{aligned} \tag{34}$$

and the ℓ_1 regularization of (32) (Markovsky & Dörfler, 2021)

$$\begin{aligned} &\text{minimize} \quad \text{over } g \quad \|w|_{\mathscr{I}_{\mathrm{tba}}} - \mathscr{H}_L(\mathscr{W}_{\mathrm{d}})|_{\mathscr{I}_{\mathrm{tba}}} g\|_{v|_{\mathscr{I}_{\mathrm{tba}}}}^2 + \lambda \|g\|_1 \\ &\text{subject to} \quad \hat{w}|_{\mathscr{I}_{\mathrm{exact}}} = w|_{\mathscr{I}_{\mathrm{exact}}}. \end{aligned} \tag{35}$$

The solution of (34) depends on the choice of the hyperparameters δ and γ. Empirical evidence suggest that the optimal value for δ is the largest one $\delta = L$. The hyperparameter γ controls the fitting accuracy versus model complexity trade-off. Since larger γ implies larger approximation error $\|w|_{\mathscr{I}_{\mathrm{tba}}} - \hat{w}|_{\mathscr{I}_{\mathrm{tba}}}\|_{v|_{\mathscr{I}_{\mathrm{tba}}}}$, the optimal choice is the smallest value, for which the rank constraint is met. It can be found by bisection (Markovsky, 2012).

The ℓ_1-norm regularization method (35) is based on the fact that, in case of exact data $\mathscr{W}_{\mathrm{d}}$, g can be chosen sparse. Namely, with $\|g\|_0$ denoting the number of nonzero elements of g, $\|g\|_0 = mL+n$. Then, the 1-norm $\|g\|_1$ can be used as a convex relaxation of $\|g\|_0$. This method is proposed in Dörfler et al. (2021a) for solving a related data-driven control problem and is used for data-driven interpolation in Markovsky and Dörfler (2021). For the numerical solution of (35) in the examples of Section 4.5, we use CVX (Grant & Boyd, 2008) and the ADMM method (Parikh & Boyd, 2014).

To summarize: problems (30)–(32) are equivalent and have the same hyperparameters (m, ℓ, n). They are the statement and data-driven reformulations of the underlying maximum-likelihood data-driven missing data estimation problem. Problem (33) is a reformulation of the inner problem in (32) as a maximum-likelihood model identification problem with the same hyperparameters (m, ℓ, n). Problems (30)–(33) are non-convex. Local optimization methods for solving them require a favorable initialization and are computationally expensive. In comparison, problems (34) and (35) are convex relaxations of the underlying problem (30). Their hyperparameters are the regularization coefficients γ and λ, and their solutions can be used as initializations for local methods solving (30). Though, the solutions of the convex relaxations (34)–(35) are also of interest in their own right and may result in favorable outcomes; see the case studies in Section 4.5. Also, there are efficient computational methods for (35) Parikh and Boyd (2014).

4.5. Numerical case studies

First, we compare the approximation methods for missing data estimation on a Monte-Carlo simulation example. Then, we compare the methods on real-data.

Simulated data

The data generating system used in this section is the benchmark example of Landau et al. (1995). It is a 4th order single-input single-output system $\mathscr{B}$ defined by a kernel representation (3) with parameter

$$\begin{aligned} R(z) = &\begin{bmatrix} -0.5067 & 0.8864 \end{bmatrix} z^0 + \begin{bmatrix} -0.2826 & -1.3161 \end{bmatrix} z^1 \\ &+ \begin{bmatrix} 0 & 1.5894 \end{bmatrix} z^2 + \begin{bmatrix} 0 & -1.4183 \end{bmatrix} z^3 + \begin{bmatrix} 0 & 1 \end{bmatrix} z^4. \end{aligned}$$

The trajectory w_{d} is generated in the errors-in-variables setup, with $\overline{w}_{\mathrm{d}} \in \mathscr{B}|_{100}$ a random trajectory of $\mathscr{B}$. The noise standard deviation is selected to match a desired noise-to-signal ratio of w_{d}. In the experiments, the noise-to-signal ratio is varied in the interval $[0, 0.1]$ (*i.e.*, up to 10% noise). The to-be-interpolated trajectory w is the step response of $\mathscr{B}$ from an input $u = w_1$ to an output $y = w_2$. The interpolation horizon is $L = 10$. Note that in this setup the two-step model-based method — (33) followed by (24) — is optimal, *i.e.*, it solves the maximum-likelihood estimation problem (30).

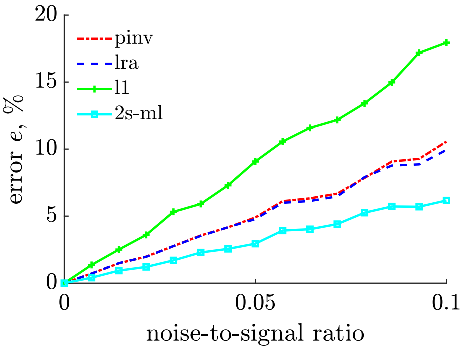

Fig. 3. The results of data-driven simulation using noisy data obtained in the errors-in-variables setting confirm empirically that the maximum-likelihood method (`2s-ml`) is statistically optimal. The performance of the two-step method with low-rank preprocessing (`lra`) is comparable with the one of the pseudo-inverse method (28) (`pinv`). Their performance is worse than the one of the maximum-likelihood method but better than the one of the ℓ_1-norm regularization method (`l1`).

As a performance metric, consider the estimation error

$$e_{\text{missing}} := \frac{\|w|_{\mathscr{I}_{\text{missing}}} - \hat{w}|_{\mathscr{I}_{\text{missing}}}\|}{\|w|_{\mathscr{I}_{\text{missing}}}\|} \, 100\%,$$

where $\hat{w}$ is the computed solution, is averaged over 100 Monte-Carlo repetitions of the experiment with different noise realizations. Fig. 3 shows that the two-step model-based method (`2s-ml`) achieves the smallest average estimation error. This is expected because it is the maximum-likelihood method for the specific simulation setup considered. The maximum-likelihood method requires nonlinear local optimization but sets a lower bound on the achievable estimation error by the other methods that are cheaper to compute albeit suboptimal in the maximum-likelihood sense. The results show that Algorithm 1, *i.e.*, the two-step method with low-rank approximation preprocessing (`lra`), is marginally better than the method based on the pseudo-inverse (28) (`pinv`). Note that the low-rank approximation preprocessing method requires knowledge of the model complexity in order to achieve the "right" complexity reduction, while the method based on the pseudo-inverse does not require any prior knowledge. Because of this, the similar performance of `pinv` and `lra` is surprising.

The ℓ_1-norm regularization method (35) (`l1`) with optimal choice of the hyperparameter $\lambda = 0.1$ gives the worst results. As shown in the next section however this is not the case when real-life data is used. Empirical evidence by Huang, Coulson et al. (2021), Markovsky and Dörfler (2021) and Wegner et al. (2021) also confirms the good performance of the ℓ_1-norm regularization for noisy data coming from nonlinear systems.

Real-data: Air passengers data benchmark

The data set used in this section is a classic time-series forecasting benchmark of Box and Jenkins (1976). It consists of 144 samples that represent the monthly totals of international airline passengers (in thousands of passengers) between 01/1949 and 12/1960. We use the first 110 samples as the given trajectory w_{d} and the remaining 34 samples as the to-be-interpolated trajectory w. From w, the first half is the given data $w|_{\mathscr{I}_{\mathrm{tba}}}$ and the second half $w|_{\mathscr{I}_{\mathrm{missing}}}$ is missing (see Fig. 4, up). For Algorithm 1, we set the parameters $m = 0$ (no inputs) and $n = 6$ (the best value obtained by trial-and-error.) The results in Table 4 (see also Fig. 4, down) show that ℓ_1-norm regularization method (35) (`l1`) with optimized value of λ achieves the best prediction. Second best is the solution based on the pseudo-inverse (28) (`pinv`).

Table 4
Performance of the data-driven missing data estimation methods on the Box–Jenkins airline passenger benchmark.

	e_{given}, %	e_{missing}, %
pinv	0	3.9168
lra	4.0384	5.2688
l1	3.3664	3.3387
2s-ml	4.0572	Fail

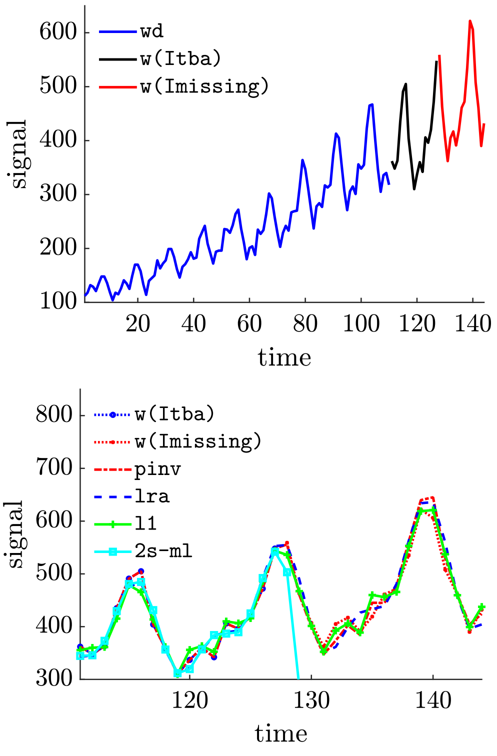

Fig. 4. Up: splitting of the data into w_d (wd), $w|_{\mathscr{I}_\mathrm{tba}}$ (w(Itba)), and $w|_{\mathscr{I}_\mathrm{missing}}$ (w(Imissing)). Down: predictions obtained by the methods. The ℓ_1-norm regularization method (35) (l1) with optimized value of λ achieves the best prediction.

The two-step model-based method 2s-ml fails (relative error above 100%) and the low-rank preprocessing also does not improve the result of (28). The poor performance of 2s-ml and lra is attributed to the fact that w_d does not satisfy a true linear time-invariant model dynamics, however, 2s-ml and lra use this as prior knowledge and enforce it in the preprocessing step. On the other hand l1 and pinv are based on a non-parametric representation which does not impose an a priori given bound on the model's complexity.

4.6. Data-driven analysis

Further system analysis problems were addressed using the data-driven representation (17). van Waarde et al. (2020) proposed stability, controllability, and stabilizability tests (for details see Section 5.3). Koch et al. (2020), Maupong et al. (2017), Romer et al. (2019) and Rosa and Jayawardhana (2021) considered data-driven dissipativity analysis, Monshizadeh (2020) considered data-driven model reduction, and Markovsky (2015) considered estimation of the DC-gain from a finite number samples of a step response. Data-driven analysis result for polynomial systems are presented in Martin and Allgöwer (2021a).

5. Data-driven control

Data-driven control methods can be loosely classified into *indirect data-driven control* approaches consisting of sequential system identification and model-based control as well as *direct data-driven control* approaches seeking an optimal decision compatible with data recorded from the system. Both approaches have a rich history, and they have received renewed interest cross-fertilized by novel methods and widespread interest in machine learning. Representative recent surveys for indirect and direct approaches are by Chiuso and Pillonetto (2019), Hewing et al. (2020), Hjalmarsson (2005), Hou and Wang (2013), Recht (2019) and Pillonetto et al. (2014), respectively.

The pros and cons of both paradigms have often been elaborated on. Whereas the indirect approach is modular and well understood, modeling and identification is cumbersome, its results are often not useful for control (due to, e.g., incompatible uncertainty quantifications), and practitioners often prefer end-to-end approaches. Direct methods promise to resolve these problems by learning control policies directly from data. However, they are often analytically and computationally less tractable and rarely apply to real-time and safety-critical systems.

The methods reviewed in this article, based on the fundamental lemma, lend themselves both for direct as well as indirect approaches. Regarding the indirect approaches, the fundamental lemma in Willems et al. (2005) has historically been developed as a foundation for subspace system identification methods based on an experiment design perspective. We refer to Markovsky et al. (2006) for a discussion on how the fundamental lemma relates to the indirect approach (i.e., system identification) and focus on direct data-driven control here.

5.1. Open-loop data-driven linear quadratic tracking

Our exposition follows up on the approaches presented in Section 4, but we impose more structure in this section. As an extension to the tracking problem (22), consider the linear quadratic (LQ) optimal tracking control problem

$$\begin{aligned}&\text{minimize over } u_\mathrm{f},\ y_\mathrm{f} \quad \sum_{t=1}^{T_\mathrm{f}} \|y_\mathrm{f}(t)-y_\mathrm{r}(t)\|_Q^2 + \|u_\mathrm{f}(t)-u_\mathrm{r}(t)\|_R^2 \\ &\text{subject to} \quad (u_\mathrm{ini}, y_\mathrm{ini}) \wedge (u_\mathrm{f}, y_\mathrm{f}) \in \mathscr{B}|_{T_\mathrm{ini}+T_\mathrm{f}}\end{aligned} \qquad (36)$$

on a finite horizon $T_\mathrm{f} > 0$, where $w_\mathrm{r} = (u_\mathrm{r}, y_\mathrm{r}) \in \mathbb{R}^{qT_\mathrm{f}}$ is a user-defined reference trajectory (not necessarily in $\mathscr{B}|_{T_\mathrm{f}}$), $w_\mathrm{f} = (u_\mathrm{f}, y_\mathrm{f}) \in \mathbb{R}^{qT_\mathrm{f}}$ is the future trajectory of length $T_\mathrm{f} \geq 1$ to be designed, and $w_\mathrm{ini} = (u_\mathrm{ini}, y_\mathrm{ini})$ is a given prefix trajectory of length $T_\mathrm{ini} \geq \ell$ setting the initial condition; see Lemma 1. 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 $Q \succeq 0$. We remark that a quadratic cost is convenient but not strictly necessary for many of the approaches reviewed in this section.

The LQ control problem (36) is an instance of errors-in-variables Kalman smoothing (21). Problem (36) is standard and can be solved by a variety of methods provided that a parametric model (typically in state–space representation) of $\mathscr{B}$ is available (Anderson & Moore, 2007). In what follows, we survey direct data-driven approaches related to the fundamental lemma and the data-driven image representation (17) of $\mathscr{B}|_{T_\mathrm{ini}+T_\mathrm{f}}$.

For simplicity, this section considers only a single data trajectory w_d and the Hankel matrix $\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})$. Extensions to multiple trajectories and mosaic-Hankel matrices are possible.

5.1.1. Data-driven approach to finite-time LQ control

Given data w_d collected offline which is persistently exciting of sufficient order, the fundamental lemma implies that the concatenated initial and future trajectory $w := w_\mathrm{ini} \wedge w_\mathrm{f} \in \mathscr{B}|_{T_\mathrm{ini}+T_\mathrm{f}}$ lies in the image of $\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})$, that is, $w = \mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(w_\mathrm{d})g$ for some g. According to

$w_{\text{ini}} = (u_{\text{ini}}, y_{\text{ini}})$ and $w_{\text{f}} = (u_{\text{f}}, y_{\text{f}})$, permute and partition the Hankel matrix as

$$\begin{bmatrix} w_{\text{ini}} \\ w_{\text{f}} \end{bmatrix} \sim \left[\begin{array}{c} u_{\text{ini}} \\ u_{\text{f}} \\ \hline y_{\text{ini}} \\ y_{\text{f}} \end{array}\right], \quad \mathscr{H}_{T_{\text{ini}}+L}(w_{\text{d}}) \sim \left[\begin{array}{c} U_{\text{p}} \\ U_{\text{f}} \\ \hline Y_{\text{p}} \\ Y_{\text{f}} \end{array}\right] = \left[\begin{array}{c} \mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}(u_{\text{d}}) \\ \hline \mathscr{H}_{T_{\text{ini}}+T}(y_{\text{d}}) \end{array}\right],$$

where $\sim$ denotes similarity under a coordinate permutation. With this notation in place, the LQ tracking control problem (36) can be posed in the equivalent data-driven formulation

$$\begin{aligned} &\text{minimize} \quad \text{over } u_{\text{f}},\ y_{\text{f}},\ g \quad \|y_{\text{f}} - y_{\text{r}}\|_Q^2 + \|u_{\text{f}} - u_{\text{r}}\|_R^2 \\ &\text{subject to} \quad \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \\ Y_{\text{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_{\text{f}} \\ y_{\text{f}} \end{bmatrix}, \end{aligned} \tag{37}$$

where (with slight abuse of notation) we redefined Q and R as blkdiag$(Q, \dots, Q)$ and blkdiag$(R, \dots, R)$, respectively.

The data-driven LQ control formulation (37) has been first presented and analyzed by Markovsky and Rapisarda (2008), and an explicit solution has been proposed. An earlier precursor and solution to data-driven LQ control based on the fundamental lemma is due to Fujisaki et al. (2004). Their approach is geometric, and the design is based on controllable and reachable subspaces which can be constructed from $\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}(w_{\text{d}})$.

If the underlying state is directly available, $Q = 0$, $T_{\text{f}} \geq n$, and a terminal condition on $y_{\text{r}}(T_{\text{f}})$ is imposed, the LQ control formulation (36) reduces to classic minimum energy control, and a similar data-driven solution has been investigated by Baggio et al. (2019) and follow-up articles (Baggio et al., 2021; Baggio & Pasqualetti, 2020). Based on numerical case studies, Baggio et al. (2019) concluded that the direct data-driven approach displays superior performance (especially for large data size and state dimension) over the explicit (model-based) minimum energy control formula invoking the controllability Gramian.

If the initial conditions are not a priori given, the data-driven LQ control problem (37) entails both estimation of an initial prefix trajectory w_{ini} (equivalent to imposing the initial condition of a latent state variable; see Lemma 1) as well as prediction and optimization of the future system behavior w_{f}. It is clean, tractable, and theoretically insightful, albeit it is not immediately clear how to extend it beyond the setting of exact data w_{d} or how to derive closed-form feedback control policies in the infinite-horizon setting. These questions will be further pursued in Sections 5.2 and 5.3 . Before that we briefly review a historic precursor to the LQ control formulation (36).

5.1.2. Subspace predictive control

Subspace predictive control (SPC) coined by Favoreel et al. (1999) is an early data-driven control approach originating from subspace system identification, which has seen plenty of theoretical developments and practical applications; see Huang and Kadali (2008) for a survey. Although SPC historically predates the fundamental lemma, it can be nicely introduced within the framework of the previous section. SPC seeks a linear relation, i.e., a matrix K, relating past and future inputs and outputs as

$$y_{\text{f}} = \underbrace{\left[K_{\text{p}} \;\middle|\; K_{\text{f}}\right]}_{=K} \left[\begin{array}{c} u_{\text{ini}} \\ y_{\text{ini}} \\ \hline u_{\text{f}} \end{array}\right]. \tag{38}$$

The multi-step predictor K can be found from data by replacing the variables $(u_{\text{ini}}, y_{\text{ini}}, u_{\text{f}}, y_{\text{f}})$ in (38) by the Hankel matrix data $(U_{\text{p}}, Y_{\text{p}}, U_{\text{f}}, Y_{\text{f}})$ and solving for K approximately in the least-square sense, that is, Huang and Kadali (2008, Section 3.4):

$$K = \arg\min_{\hat{K}} \left\| Y_{\text{f}} - \hat{K} \cdot \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \end{bmatrix} \right\|_F = Y_{\text{f}} \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \end{bmatrix}^{\dagger}, \tag{39}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and uniqueness holds under full rank conditions descending from, e.g., the fundamental lemma. For exact data, (38)–(39) is an ARX model with rank$(K_{\text{p}}) = n$ assuring LTI behavior of desired complexity and a lower block-triangular zero pattern of K_{f} assuring causality. For inexact data, LTI behavior of desired complexity is promoted by low-rank approximation (typically, via singular-value thresholding of K_{p}) (Favoreel et al., 1999). By heuristically thresholding K_{f} towards a block-triangular zero pattern one aims to gain causality (Huang & Kadali, 2008, Remark 10.1).

These steps bring the linear relation (38) half-way towards an LTI model. Though a model has further structure, e.g., K_{f} is Toeplitz, and the entries of K_{p} and K_{f} are coupled. Nevertheless, the linear relation (38)–(39) without further post-processing has demonstrated excellent performance as a data-driven predictor employed in receding-horizon predictive control across various case studies; see Huang and Kadali (2008), Lu et al. (2014), Vajpayee et al. (2017) and Zeng et al. (2010) for an overview.

Connections between SPC and data-driven LQ control

The close connections between the SPC predictor (38)–(39) and the direct data-driven LQ control problem (37) have been remarked upon a few times (Dörfler et al., 2021a; Fiedler & Lucia, 2021; Huang et al., 2019), and we summarize them below.

Observe that the variable g can be eliminated from the constraint of (37) as $y_{\text{f}} = Y_{\text{f}} g$, where g is any solution to the remaining constraint equations. Whereas the solution g is not necessarily unique, the resulting output y_{f} is unique (Markovsky & Rapisarda, 2008, Proposition 1). One choice is the associated least norm-solution, that is, $y_{\text{f}} = Y_{\text{f}} g^{\star}$ where

$$\begin{aligned} g^{\star} = \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \end{bmatrix}^{\dagger} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_{\text{f}} \end{bmatrix} = \ &\arg\min \quad \text{over } g \quad \|g\|_2 \\ &\text{subject to} \quad \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_{\text{f}} \end{bmatrix}. \end{aligned} \tag{40}$$

With this reformulation and elimination of g the direct data-driven LQ control problem (37) reduces to

$$\begin{aligned} &\text{minimize} \quad \text{over } u_{\text{f}},\ y_{\text{f}} \quad \|y_{\text{f}} - y_{\text{r}}\|_Q^2 + \|u_{\text{f}} - u_{\text{r}}\|_R^2 \\ &\text{subject to} \quad y_{\text{f}} = Y_{\text{f}} \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \end{bmatrix}^{\dagger} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_{\text{f}} \end{bmatrix}, \end{aligned} \tag{41}$$

that is, we recover the multi-step SPC predictor (38)–(39).

Observe that the reformulation of the direct data-driven LQ control problem (37) towards SPC was only possible since g was unconstrained, unpenalized, and any solution results in the same output — in case of exact data. In case of inexact data, the reformulation (40) suggests a regularization of the LQ problem (37) with $\|g\|_2$ to filter out noise akin to least squares (39). We will further pursue this line of ideas in the next section.

5.2. Data-enabled predictive control

For deterministic LTI systems the direct data-driven LQ tracking control (37) can be implemented *at face value* in a receding-horizon predictive control fashion; see Yang and Li (2013) or the related SPC literature (Huang & Kadali, 2008). Indeed, in this case, it can be shown that the data-driven LQ problem (37) is equivalent to a model-based predictive control (MPC) formulation (Coulson et al., 2019a, 2020).

When departing from deterministic LTI systems and exact data, it is tempting to opt for a certainty-equivalence implementation, that is, to implement the control as in (37) despite not satisfying the assumptions. However, the latter approach fails. This can be intuitively understood from the perspective of dictionary learning. The columns of the Hankel matrix $\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}(w_{\text{d}})$ serve as a library of trajectories, and the LQ problem (37) linearly combines these trajectories to synthesize the optimal control trajectory. However, a superposition of trajectories from

$\mathscr{B}|_{T_{\mathrm{ini}}+T_{\mathrm{f}}}$ is again a valid trajectory of $\mathscr{B}|_{T_{\mathrm{ini}}+T_{\mathrm{f}}}$ only for linear systems. Even in the linear stochastic case a superposition of trajectories does not generally preserve the noise statistics, e.g., a linear combination of Gaussian random variables with identical variance equals another Gaussian random variable though with a generally different variance. Even more detrimental: a Hankel matrix $\mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(w_{\mathrm{d}})$ built from noisy data will likely have full rank, not reveal an LTI behavior of bounded complexity, and any optimal control trajectory w is feasible for (37), that is, the predicted optimal trajectory can be arbitrarily optimistic and non-realizable when applied to the real system.

Aside from the above issues related to the data w_{d} collected *offline*, the data $w_{\mathrm{ini}} = (u_{\mathrm{ini}}, y_{\mathrm{ini}})$ collected *online* (before implementing an instance of the optimal control (37)) is typically noise-corrupted as well which leads to feasibility issues and further deterioration of the realized control performance.

For these reasons the certainty-equivalence approach has to be replaced by a robust one. Below we review *Data-EnablEd Predictive Control* (known by its acronym DeePC) coined by Coulson et al. (2019a) as a robustified receding-horizon implementation of the direct data-driven LQ tracking control (37).

5.2.1. Robustified formulation of the direct data-driven LQ optimal control problem & DeePC

As previously discussed, the need for robustification of the direct data-driven LQ problem (37) is two-fold. First, note that when implementing (37) in receding-horizon, the data $w_{\mathrm{ini}} = (u_{\mathrm{ini}}, y_{\mathrm{ini}})$ is measured and repeatedly updated online. In case of inexact data, due to measurement noise and input/output disturbances, the constraint equations $U_{\mathrm{p}} g = u_{\mathrm{ini}}$ and $Y_{\mathrm{p}} g = y_{\mathrm{ini}}$ determining the initial behavior may not be feasible. As a remedy, DeePC opts for a moving-horizon least-error estimation (Rawlings et al., 2017) and softens these constraints as

$$\begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \end{bmatrix} g = \begin{bmatrix} u_{\mathrm{ini}} + \sigma_{u_{\mathrm{ini}}} \\ y_{\mathrm{ini}} + \sigma_{y_{\mathrm{ini}}} \end{bmatrix}, \tag{42}$$

where $\sigma_{u_{\mathrm{ini}}}$ and $\sigma_{y_{\mathrm{ini}}}$ are slack variables penalized in the cost.

Second, aside from the above *additive* uncertainty, the data-driven LQ problem (37) is also subject to *multiplicative* uncertainty, since the data-matrices U_{p}, Y_{p}, U_{f}, and Y_{f} are also subject to noise. This noise can be mitigated offline by pre-processing the trajectory library (e.g., by seeking a low-rank approximation of $\mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(w_{\mathrm{d}})$ as in Section 4.4), but in the spirit of direct data-driven control – seeking an online decision based on raw data – DeePC opts for a regularization of the LQ problem (37). In particular, a nonnegative term $h(g)$ is added to the cost function. This regularization term will be justified later in Section 5.2.2, but the attentive reader may recall from Section 4.4 that $h(g) = \|g\|_1$ corresponds to a convex relaxation of a low-rank approximation denoising scheme, and $\|g\|_2$ is connected to a pre-conditioning of the predictor à la SPC in (40).

A third minor – yet practicably important – modification is to augment the data-driven LQ problem (37) with input and output constraints $u_{\mathrm{f}} \in \mathscr{U}$ and $y_{\mathrm{f}} \in \mathscr{Y}$, respectively. These can account for, e.g., input saturation, operational limits, or terminal constraints needed for closed-loop stability of the predictive control scheme (Borrelli et al., 2017; Rawlings et al., 2017).

These three modifications give rise to the DeePC problem

$$\begin{aligned}
\text{minimize} \quad & \text{over } u_{\mathrm{f}},\, y_{\mathrm{f}},\, g,\, \sigma_{u_{\mathrm{ini}}},\, \sigma_{y_{\mathrm{ini}}} \quad \|y_{\mathrm{f}} - y_{\mathrm{r}}\|_Q^2 + \|u_{\mathrm{f}} - u_{\mathrm{r}}\|_R^2 \\
& + \lambda_{u_{\mathrm{ini}}} \|\sigma_{u_{\mathrm{ini}}}\|_2^2 + \lambda_{y_{\mathrm{ini}}} \|\sigma_{y_{\mathrm{ini}}}\|_2^2 + \lambda_g \cdot h(g) \\
\text{subject to} \quad & \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{\mathrm{ini}} + \sigma_{u_{\mathrm{ini}}} \\ y_{\mathrm{ini}} + \sigma_{y_{\mathrm{ini}}} \\ u_{\mathrm{f}} \\ y_{\mathrm{f}} \end{bmatrix} \text{ and } (u_{\mathrm{f}}, y_{\mathrm{f}}) \in \mathscr{U} \times \mathscr{Y},
\end{aligned} \tag{43}$$

where $\lambda_{u_{\mathrm{ini}}}$, $\lambda_{y_{\mathrm{ini}}}$, and λ_g are nonnegative scalar regularization coefficients (hyperparameters). Many variations of the estimation penalty $\lambda_{u_{\mathrm{ini}}} \|\sigma_{u_{\mathrm{ini}}}\|_2^2 + \lambda_{y_{\mathrm{ini}}} \|\sigma_{y_{\mathrm{ini}}}\|_2^2$ are conceivable, e.g., choosing norms weighted by inverse noise covariances, disregarding the penalty on $\sigma_{u_{\mathrm{ini}}}$ in absence of input noise, or removing the squares in the spirit of exact penalization, i.e., for sufficiently large $(\lambda_{u_{\mathrm{ini}}}, \lambda_{y_{\mathrm{ini}}})$ the slack variables $(\sigma_{u_{\mathrm{ini}}}, \sigma_{y_{\mathrm{ini}}})$ take a non-zero value only if the constraints are infeasible.

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

$$\begin{aligned}
\text{minimize} \quad & \text{over } g \quad \|Y_{\mathrm{f}} g - y_{\mathrm{r}}\|_Q^2 + \|U_{\mathrm{f}} g - u_{\mathrm{r}}\|_R^2 \\
& + \lambda_{y_{\mathrm{ini}}} \|Y_{\mathrm{p}} g - y_{\mathrm{ini}}\|_2^2 + \lambda_{u_{\mathrm{ini}}} \|U_{\mathrm{p}} g - u_{\mathrm{ini}}\|_2^2 + \lambda_g \cdot h(g) \\
\text{subject to} \quad & (U_{\mathrm{f}} g, Y_{\mathrm{f}} g) \in \mathscr{U} \times \mathscr{Y}\,.
\end{aligned} \tag{44}$$

In fact, in absence of constraints, (44) takes the form of a regularized regression problem

$$\text{minimize over } g \quad \|\mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(w_{\mathrm{d}}) g - w_{\mathrm{r,ini}}\|_P^2 + \lambda_g \cdot h(g), \tag{45}$$

where P is the block-diagonal matrix $\mathrm{blkdiag}(\lambda_{u_{\mathrm{ini}}} I, \lambda_{y_{\mathrm{ini}}} I, R, Q)$ and $w_{\mathrm{r,ini}} = (u_{\mathrm{ini}}, y_{\mathrm{ini}}, u_{\mathrm{r}}, y_{\mathrm{r}})$. The latter compact formulation does not only provide a regression perspective on the DeePC problem, but also motivates the use of Bayesian, non-parametric, or robust regression methods to approach and extend the DeePC problem formulation (45).

5.2.2. Robustification of DeePC by means of regularization

The regularization term $h(g)$ in (43)–(45) is needed to robustify the optimal control design in case of inexact data w_{d} arising from possibly non-deterministic and nonlinear processes. The regularizations have first been proposed heuristically by Coulson et al. (2019a) before being constructively derived. Different assumptions on the data lead to different regularizers. In what follows, we briefly review five different variations.

(1) Regularization derived from pre-processing

In case of inexact data, the matrix $\mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(w_{\mathrm{d}})$ will generically not have the desired rank $m(T_{\mathrm{ini}}+T_{\mathrm{f}})+n$ and will not reveal an LTI behavior of desired complexity. As in Section 4.4, the noisy data matrix can be pre-processed via structured low-rank approximation. Formally, this can be posed as a *bi-level optimization problem*: namely, solve the optimal control problem subject to pre-processing of the data matrix as in (33):

$$\begin{aligned}
\text{minimize} \quad & \text{over } g \quad \|\mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(\widehat{w}_{\mathrm{d}}^{\star}) g - w_{\mathrm{r,ini}}\|_P^2 \\
\text{subject to} \quad & \widehat{w}_{\mathrm{d}}^{\star} \in \arg\min \ \text{over } \widehat{w}_{\mathrm{d}} \text{ and } \widehat{\mathscr{B}} \quad \|w_{\mathrm{d}} - \widehat{w}_{\mathrm{d}}\| \\
& \qquad \text{subject to} \quad \widehat{w}_{\mathrm{d}} \in \widehat{\mathscr{B}}|_T \text{ and } \widehat{\mathscr{B}} \in \mathscr{L}_c^q
\end{aligned} \tag{46}$$

This non-convex bi-level decision making problem can be formally reduced and convexified as in (35) leading to the direct DeePC formulation (45) with an ℓ_1-norm regularization $h(g) = \|g\|_1$; see Dörfler et al. (2021a, Theorem 4.6) for details.

(2) Regularization derived from least-square identification

As a second source of regularization, consider solving the optimal control problem (37) (neglecting constraints and noisy estimation for simplicity) with an ARX predictor as in SPC (38), where the multi-step predictor K is found by ordinary least squares as in (39). This procedure can be formally posed again as a non-convex bi-level decision making problem:

$$\begin{aligned}
\text{minimize} \quad & \text{over } u_{\mathrm{f}},\, y_{\mathrm{f}} \quad \|y_{\mathrm{f}} - y_{\mathrm{r}}\|_Q^2 + \|u_{\mathrm{f}} - u_{\mathrm{r}}\|_R^2 \\
\text{subject to} \quad & y_{\mathrm{f}} = K^{\star} \left[\begin{array}{c} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ \hline u_{\mathrm{f}} \end{array}\right] \\
& K^{\star} = \arg\min \ \text{over } K \quad \left\| Y_{\mathrm{f}} - \underbrace{\left[\begin{array}{c|c} K_p & K_f \end{array}\right]}_{=K} \left[\begin{array}{c} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ \hline U_{\mathrm{f}} \end{array}\right] \right\|_F \\
& \text{subject to} \quad \mathrm{rank}(K_p) = n \\
& \qquad\qquad\quad \text{and} \quad K_f \text{ lower-block triangular}
\end{aligned} \tag{47}$$

Here, the rank constraint on K_p promotes an LTI behavior of desired complexity, and the lower-block triangular structure of K_f assures causality, as discussed after (39). When dropping these constraints and re-parameterizing the least-square criterion by a least-norm problem as in (40), the DeePC formulation (43) can be derived as a convex relaxation to (47) with regularizer

$$h(g) = \left\| \left(I - \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix}^\dagger \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} \right) g \right\|, \tag{48}$$

where $\| \cdot \|$ is any norm; see Dörfler et al. (2021a, Theorem 4.5) for details. This projection-based regularizer assures that a particular least-norm solution g is singled out corresponding to the least-square criterion in (40). Finally, we note that the projection-based regularizer (48) is consistent, i.e., the solution of the LQ problem (37) also solves the regularized problem (43) in case of exact data w_d. In comparison, mere norm-based regularizers $h(g) = \|g\|$ are not consistent and bias the solution.

Synopsis: The above two direct and regularized data-driven control approaches are due to reducing and convexifying the indirect "first pre-process/identify and then control" problems. In either case, the magnitude of the regularization coefficient λ_g ensures to which extent the inner pre-processing/identification problems are (approximately) enforced. However, unlike (46) or (47) no projection on the class of LTI models of desired complexity is enforced. As a result, 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 a direct DeePC formulation; see Dörfler et al. (2021a) for a discussion. Further trade-offs between the direct and indirect approaches are discussed by Krishnan and Pasqualetti (2021) concluding that either approach can be superior depending on prediction horizon, state dimension, noise level, and size of the data set.

(3) Regularizations derived from robust optimization

An entirely different route towards regularization can be derived by robustifying the regression-based DeePC formulation (45) (without constraints and regularization) as in related robustified regression problems (Bertsimas & Copenhaver, 2018; El Ghaoui & Lebret, 1997; Xu et al., 2010)

$$\text{minimize over } g \quad \text{maximize over } \hat{w}_\mathrm{d} \in \mathbb{W}(w_\mathrm{d}) \quad \|\mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(\hat{w}_\mathrm{d})g - w_\mathrm{r,ini}\|_P^2\,, \tag{49}$$

where $\mathbb{W}(w_\mathrm{d})$ is an uncertainty set typically centered at the collected offline data w_d. Huang, Jianzhe et al. (2021) and Huang, Zhen et al. (2021) consider different structured and unstructured uncertainty sets ranging from mere norm balls, over interval-valued and column-wise uncertainties (relevant for a trajectory matrix structure), to uncertainties with Hankel structure. For each of these Huang, Jianzhe et al. (2021) and Huang, Zhen et al. (2021) propose tractable reformulations, many of which take the form (45) with norm-based regularization terms $h(g)$. Moreover, Huang, Jianzhe et al. (2021) also consider the case of robustified constraints and provide bounds on the realized system performance.

(4) Regularizations derived from distributional robustness

A similar (albeit stochastic) perspective leading to regularization is due to distributional robustness (Kuhn et al., 2019). Problem (44) (without regularization term) can be abstracted as

$$\text{minimize} \quad \text{over } g \in \mathscr{G}(w_\mathrm{d}) \quad f(w_\mathrm{d}, g)\,, \tag{50}$$

where $\mathscr{G}(w_\mathrm{d})$ and $f(w_\mathrm{d}, g))$ denote the constraint set and objective of (44), respectively. Since the data w_d has arisen from a stochastic process, one may equivalently rewrite (50) as

$$\text{minimize} \quad \text{over } g \in \mathscr{G}(w_\mathrm{d}) \quad \mathbb{E}_{\hat{w}_\mathrm{d}\sim\hat{\mathbb{P}}}\left[f(\hat{w}_\mathrm{d}, g)\right]\,, \tag{51}$$

where $\hat{\mathbb{P}}$ is the associated empirical distribution built using the measured data w_d, i.e., the measure of $\hat{w}_\mathrm{d}$ concentrates on w_d.

If the solution of the *sample-average problem* (51) is implemented on the real system, one suffers an out-of-sample loss since the true data-generating distribution $\mathbb{P}$ is due to some (possibly nonlinear, non-stationary, non-Gaussian) stochastic process that is only poorly represented by the samples $\hat{\mathbb{P}}$. To be robust against such processes, Coulson et al. (2019b, 2020) propose the distributionally robust DeePC formulation

$$\inf_{g\in\mathscr{G}} \sup_{\mathbb{Q}\in\mathbb{B}_\epsilon^p(\hat{\mathbb{P}})} \mathbb{E}_{\hat{w}_\mathrm{d}\sim\mathbb{Q}}\left[f(\hat{w}_\mathrm{d}, g)\right]\,, \tag{52}$$

where the ambiguity $\mathbb{B}_\epsilon^p(\hat{\mathbb{P}})$ is a Wasserstein ball of radius $\epsilon > 0$, centered at $\hat{\mathbb{P}}$, and with metric induced by the ℓ_p-norm. One can show that, under integrability conditions and for a Lipschitz objective, the distributionally robust formulation (52) is equivalent to the regularized DeePC (44) with λ_g being ϵ times the Lipschitz constant of the cost and with regularizer $h(g) = \|g\|_p^\star$, where $\| \cdot \|_p^\star$ is the dual norm of that one used to construct the Wasserstein ball (Coulson et al., 2020, Theorem 4.1). For example, safeguarding against uncertainty in ℓ_∞-norm in the space of trajectories is equivalent to ℓ_1-norm regularization.

The same methods can also be applied to distributionally robustify stochastic formulations of constraints (Coulson et al., 2020). Furthermore, data compression and sample-complexity results are in Coulson et al. (2020) and Fabiani and Goulart (2020).

(5) Regularization related to robust control

Xue and Matni (2020) propose a data-driven formulation of the system level synthesis (SLS) subspace constraint (Anderson et al., 2019) (parameterizing the admissible closed-loop responses) by means of the non-parametric representation (17) and assuming full state measurements. The resulting robust LQ formulation (with bounded adversarial disturbances on the data matrix) results again in a norm-based regularization (Xue & Matni, 2020, eq. (3.9)). Building on this connection to SLS, Lian and Jones (2021b) provide an extension to a class of uncertain LTI systems and a DeePC formulation with disturbance-affine feedback. Furthermore, Furieri et al. (2021a, 2021b) extend these works towards safety constraints, measurement and process noise, and partial observations (i.e., output feedback) by deriving a data-driven reformulation of the recently proposed input–output parametrization (Furieri et al., 2019). The authors also address robust and constrained linear quadratic Gaussian (LQG) design, provide a tractable upper bound, and a suboptimality certificate with respect to the ground-truth LQG control.

Aside from regularization, we also mention the following methods seeking a robust DeePC formulation. Xu et al. (2021) consider measurement noise within ellipsoidal uncertainty sets, characterize noise sequences consistent with the data, and transform the robust control problem to a semidefinite program via the S-Lemma. Yin et al. (2020a, 2020b) propose a maximum likelihood framework to DeePC under Gaussian noise: namely, starting from (37) a vector g is sought that maximizes the likelihood of observing both the predicted and the measured output trajectories y_f and y_ini, respectively. The resulting formulation is amenable to a sequential quadratic programming approach.

Last, we remark that all of the above approaches towards robustified DeePC empirically show excellent closed-loop performance and may outperform each other depending on the specific problem scenario, noise characteristics, etc., as demonstrated in various case studies; see also Section 5.2.4. In fact, regularization is a key aspect in the formal closed-loop stability and robustness analysis by Berberich et al. (2021b) (see also Section 5.2.3) for noisy data or when interconnecting DeePC with a nonlinear system. Note that certain regularizers can be justified in multiple ways, e.g., the ℓ_1-norm regularizer might arise due to low-rank pre-processing, a robust regression formulation, or a distributional robust formulation. Finally, both theoretic and empirical results show that the performance of some robustifications is superior when choosing particular data matrix structures, e.g., Page and trajectory matrices (15)–(16) with independent columns instead of Hankel matrices (Coulson et al., 2020; Huang, Coulson et al., 2021; Huang, Jianzhe et al., 2021).

5.2.3. Closed-loop receding-horizon and recursive implementations: certificates and extensions

The previous subsections have focused mostly on the optimization formulation of DeePC with exception of Furieri et al. (2021a, 2021b), Huang, Jianzhe et al. (2021) and Xue and Matni (2020) that also certify the realized open-loop control performance optimizing over input vectors and affine control policies, respectively.

There have also been various approaches towards certifying the closed-loop behavior when implementing DeePC as receding-horizon control. Notable are the sequence of papers (Berberich et al., 2020a, 2020b, 2021c, 2021d; Bongard et al., 2021). The initial work by Berberich et al. (2021b) provides closed-loop stability and robustness guarantees (in the sense of practical exponential stability) of DeePC (43) with ridge regularizer $h(g) = \|g\|_2^2$ and terminal equilibrium constraints. Later articles extend and complement this work towards robust output constraint satisfaction Berberich et al. (2020b), time-varying Refs. Berberich et al. (2020a), less restrictive terminal ingredients (Berberich et al., 2021d), implementations without terminal ingredients (Bongard et al., 2021), and linear tracking control for nonlinear systems with online data adaptation (Berberich et al., 2021c). Independently, Berberich et al. (2021c) also provide a version of the fundamental lemma for affine systems resulting from linearizing nonlinear systems. In comparison to other nonlinear fundamental lemma extensions (briefly reviewed after Lemma 2) which gave rise to DeePC implementations, Berberich et al. (2021c) also certify the resulting nonlinear closed-loop properties.

Alpago et al. (2020) depart from the receding-horizon estimation (42) in DeePC towards a recursive Kalman filtering approach using the parametric solution of the optimization problem (45) to construct a hidden state. As an independent side note, the parametric solution takes the form of a piece-wise linear affine feedback policy (compared to standard linear feedback policies) which sheds further light on the remarkable performance of DeePC when applied to nonlinear systems.

Finally, recently Baros et al. (2020), Bianchin et al. (2021) and Nonhoff and Müller (2021) applied online feedback optimization (i.e., iterative algorithms in feedback with a system (Hauswirth et al., 2021)) to steer an LTI system characterized by the fundamental lemma. In a similar algorithmic spirit, Allibhoy and Cortés (2020) and Alexandru et al. (2021) consider networked and distributed DeePC implementations, respectively.

5.2.4. DeePC: implementations and tuning recommendations

At its core, DeePC is a method for deterministic LTI systems based on super-imposing trajectories from a library. It is due to the various robustifications reviewed in this section that make DeePC "work" for nonlinear and stochastic systems, as spectacularly showcased by many experimental and numerical case studies. Below we summarize some practical validations and recommendations for tuning the DeePC hyperparameters.

Notable experimental and computational DeePC case studies

Coulson et al. (2019a, 2019b, 2020) use an aerial robotics simulation case study via the DeePC method. Elokda et al. (2019) experimentally implemented this case study and for the first time demonstrated the performance, robustness, and real-time implementability of DeePC. A video of a quadcopter successfully tracking step commands and a figure 8 trajectory can be found here: https://polybox.ethz.ch/index.php/s/ZHacWoJbxQlHDTz. Further within the realm of robotics, DeePC has also been experimentally implemented to swing up a laboratory pendulum (Tischhauser et al., 2019) as well as to control a 12 ton autonomous walking excavator (Wegner et al., 2021); see Fig. 5 for an illustration. These two case studies are strongly nonlinear. While DeePC succeeds in meeting the specifications, these two case studies also reveal the limitations of the method and suggest an adaptive DeePC method to provide tracking for strongly nonlinear systems.

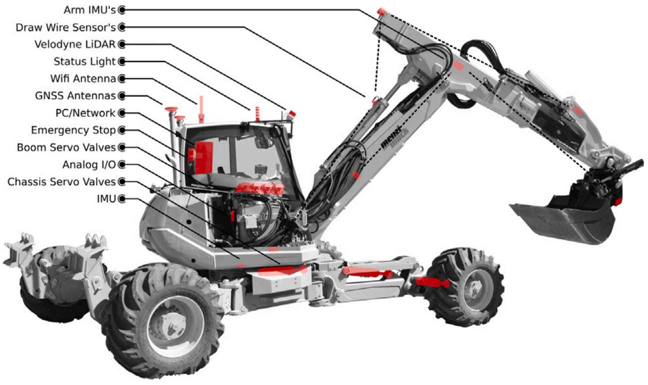

Fig. 5. The highly customized Menzi Muck M545 12 ton autonomous walking excavator from the HEAP (Hydraulic Excavator for an Autonomous Purpose) project (Jud et al., 2021) served as a demonstration platform for DeePC.

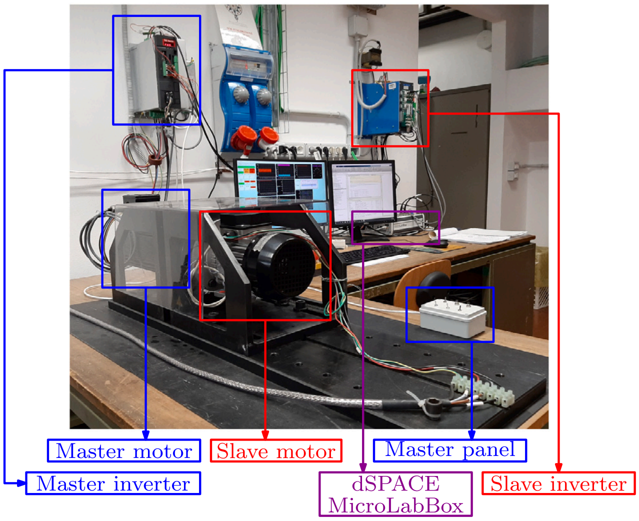

Fig. 6. Carlet et al. (2021) used a synchronous motor drive test bench at the EDLab Padova as a demonstration platform for DeePC and for comparisons to SPC and certainty-equivalence control based on an identified model.

On the power systems and electronics side, motivated by an initial numerical case study (Huang et al., 2019), DeePC has been successfully experimentally implemented on grid-connected power converters (Huang, Zhen et al., 2021) and synchronous motor drives (Carlet et al., 2021, 2020); see Fig. 6 for an illustration of the laboratory setup. Further, Huang, Coulson et al. (2021) provide a decentralized DeePC implementation for power system oscillation damping in a large-scale numerical case study, which has also been successfully replicated by R&D groups on industrial simulators. These studies showed that implementing DeePC on microcontrollers is feasible albeit computationally challenging. Further within the realm of energy, Lian, Shi et al. (2021) and Schwarz et al. (2019) study numerical and experimental implementations for building automation.

Finally, Berberich et al. (2021a) successfully applied DeePC to a nonlinear laboratory four-tank process. We note that both Berberich et al. (2021a) and Lian, Shi et al. (2021) consider adaptive implementations updating the data online.

Many of the above case studies are safety-critical systems with complex dynamics for which constraint satisfaction, closed-loop stability, and real-time computation are essential. Remarkably, the very same DeePC method with minor adjustments has succeeded in all case studies from different areas. Next we visit the crucial hyperparameters for DeePC tuning.

DeePC tuning recommendations

We now turn towards tuning recommendations for the DeePC hyperparameters. Most hyperparameters also occur in model-based MPC such as the horizon T_{f}, cost matrices Q and R, as well as transient and terminal constraints $\mathscr{U}$ and $\mathscr{Y}$. We refer to Borrelli et al. (2017) and Rawlings et al. (2017) for standard tuning recommendations, such as a sufficiently long horizon T_{f} for closed-loop stability. The hyperparameters $(\lambda_{u_{\mathrm{ini}}}, \lambda_{y_{\mathrm{ini}}})$ are as in moving-horizon estimation (Rawlings et al., 2017) or general Kalman filtering and account for (assumed) noise covariances. Generally, $(\lambda_{u_{\mathrm{ini}}}, \lambda_{y_{\mathrm{ini}}})$ should be chosen sufficiently large. This leaves us with the estimation horizon T_{ini}, data length T, regularization function $h(g)$ and coefficient λ_g as unique DeePC hyperparameters. While each case study is different, the following favorable tuning recommendations have emerged.

First, T_{ini} controls the (presumed) model complexity. Namely, Lemma 1 requires the initial horizon T_{ini} to be longer than the lag $\mathbf{l}(\mathscr{B})$ which by (2) is again bounded by the order $\mathbf{n}(\mathscr{B})$ – both of which are unknown in a data-driven setting. It proved useful to choose T_{ini} simply sufficiently large: generally, the realized closed-loop performance monotonically improves but does increasingly less so after a certain threshold. For nonlinear systems this threshold is larger than the state-dimension confirming the intuition that a higher-order LTI model can better explain the data. Second, the length T of the data time series w_{d} has to be sufficiently long to assure persistency of excitation; see the fundamental lemma. The analytic lower bound for T depends on $\mathbf{n}(\mathscr{B})$ which is generally unknown. Similar to T_{ini}, a sufficiently large T proves beneficial. The second author has had good experiences with choosing T so that the data matrix $\mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}}(w_{\mathrm{d}})$ is square. Third and finally, all regularizers $h(g)$ perform well after tuning λ_g, and they can also be combined. As discussed previously, the projection-based regularizer (40) ensures consistency whereas norm-based regularizers robustify the control at the cost of a bias. Concerning the coefficient λ_g, most case studies approximately display a convex behavior: the closed-loop 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. While the desired λ_g can often be theoretically characterized (e.g., by the size of the uncertainty set in (49) or (52)), it remains a design parameter practically found by increasing λ_g logarithmically from a small value until the realized cost increases again.

5.2.5. Conceptually related approaches of relevance

We have throughout our review pointed to closely related data-driven predictive control formulations based on the fundamental lemma such as SPC (Favoreel et al., 1999; Huang & Kadali, 2008). There are other less closely but certainly conceptually related approaches that we briefly discuss below.

One may take the vantage point that, given a single offline data trajectory w_{d}, DeePC is able to synthesize all admissible future trajectories for predictive control. A similar perspective is taken by dynamic matrix control (DMC) (Cutler & Ramaker, 1980; Garcia et al., 1989), a historic precursor to MPC originating from industry. DMC is a predictive control method that designs future system trajectories based on a previously recorded zero-initial condition step response. Although DMC has many limitations (Lundström et al., 1995), it motivates data-driven predictive control based on a single data trajectory.

Another perspective is that of dictionary learning. The connection to DeePC is most obvious when using the trajectory matrix structure (15) and looking at the regression formulation (45). DeePC linearly combines trajectories from this dictionary to synthesize the optimal control trajectory, and the role of regularization is to avoid overfitting. A conceptually related ℓ_1-regularized dictionary learning predictive control approach has been presented by Kaiser et al. (2018). Likewise, Salvador et al. (2019) use affine combinations of stored trajectories in order to achieve offset-free tracking. Further, in the field of robotics the idea of combining or concatenating stored motion primitives (i.e., trajectories) has often been exploited for motion planning (Frazzoli et al., 2005) or predictive control (Gray et al., 2012).

Finally, the geometric approach to optimal control (Marro et al., 2002) relying on controlled and conditioned invariant subspaces is by nature coordinate-free and can also be approached in a representation-free setting based on the fundamental lemma, as demonstrated by Fujisaki et al. (2004).

5.3. Data-driven design of explicit feedback control policies

The identifiability condition (17) and the fundamental lemma provide a non-parametric representation of the restricted behavior $\mathscr{B}|_L$ based on raw data, which immediately lends itself as predictor and estimator for finite horizon feedforward and receding-horizon control, as presented in Sections 5.1–5.2. By means of the weaving lemma (Markovsky et al., 2005, Lemma 3), this predictor can in certain instances be extended to an infinite-horizon setting, but is more conceptual than practically useful in case of noisy data, and it is not immediately clear how to obtain a recursive model as well as an explicit infinite-horizon feedback control law, e.g., the setting of the linear quadratic regulator (LQR). In a state–space setting, the two articles (De Persis & Tesi, 2019; van Waarde et al., 2020) provided equally simple as well as ingenuous approaches on how to parametrize an explicit state feedback design by means of state–space data matrices and subspace relations amongst them.

5.3.1. Prototypical LTI stabilization & LQR problems

Consider a controllable input/state system as in (11)

$$\left\{ (u,x) \in (\mathbb{R}^{m+n})^{\mathbb{N}} \mid \sigma x = Ax + Bu \right\}, \tag{53}$$

where $u(t) \in \mathbb{R}^m$, and the state $x(t) \in \mathbb{R}^n$ is explicitly available as measurement. We will later comment on possible extensions if only outputs are available.

To illustrate the utility of different approaches, we consider two prototypical control problems, namely using state feedback $u = Kx$ for either model-based stabilization

$$\underset{K}{\text{find}} \quad A + BK \text{ is Schur stable}$$

or infinite-horizon LQR optimal control

$$\begin{array}{ll} \text{minimize} \quad \text{over } u & \displaystyle\sum_{t=1}^{\infty} \|x(t)\|_Q^2 + \|u(t)\|_R^2 \\ \text{subject to} & \sigma x = Ax + Bu, \end{array}$$

where $Q \succeq 0$, $R \succ 0$, and $(Q^{1/2}, A)$ is detectable. By means of the familiar Lyapunov and Gramian matrices, the stabilization and LQR problems can be parameterized as

$$\underset{K,P\succ 0}{\text{find}} \quad (A+BK)P(A+BK)^\top - P \prec 0 \tag{54}$$

and anticipating the solution $u = Kx$ as

$$\begin{array}{ll} \text{minimize} \quad \text{over } K, P \succeq I & \operatorname{trace}\left(QP + K^\top RKP\right) \\ \text{subject to} & (A+BK)P(A+BK)^\top - P + I \preceq 0, \end{array} \tag{55}$$

respectively. The LQR problem (55) admits many different parameterizations, and the proposed form (55) can be turned into a convex semidefinite program after a change of variables $Y = KP$; see Feron et al. (1992) for the continuous-time case.

Observe that both problems (54) and (55) are semidefinite optimization (respectively, feasibility) problems parameterized in terms of the closed-loop matrix $A + BK$. Likewise, many other instances of robust and optimal control can be formulated and parameterized similarly; see e.g. Scherer and Weiland (2000). The stabilization (54) and LQR (55) problems serve as running examples for the developments in this section.

5.3.2. Subspace relations in state–space data

Consider time series of length L (respectively, $L+1$) of inputs and states recorded from the LTI system (53):

$$\begin{aligned} U &:= \begin{bmatrix} u(0) & u(1) & \dots & u(L-1) \end{bmatrix}, \\ X &:= \begin{bmatrix} x(0) & x(1) & \dots & x(L) \end{bmatrix}. \end{aligned}$$

We partition the state data into predecessor and successor states:

$$\begin{aligned} X_- &:= \begin{bmatrix} x(0) & x(1) & \dots & x(L-1) \end{bmatrix}, \\ X_+ &:= \begin{bmatrix} x(1) & x(2) & \dots & x(L) \end{bmatrix} = \sigma X_- . \end{aligned}$$

Since these time series satisfy the dynamics (53), we have that

$$X_+ = \begin{bmatrix} B & A \end{bmatrix} \begin{bmatrix} U \\ X_- \end{bmatrix}. \tag{56}$$

Recall Corollary 3 of the fundamental lemma: namely, if $(u(0), u(1), \dots, u(L-1))$ is persistently exciting of order $n+1$, then the input-state data matrix has full row rank:

$$\text{rank} \begin{bmatrix} U \\ X_- \end{bmatrix} = n + m . \tag{57}$$

The rank condition (57) ensures that the pseudo-inverse of the input-state data matrix is a right inverse, that is, the measurement data Eq. (56) can be solved for (B, A). Different approaches towards stabilization (54) and LQR (55) may now be pursued from the two subspace relations (56) and (57).

5.3.3. Least-square identification of a parametric state–space model and certainty-equivalence control

Recall that the conventional approach to data-driven control is indirect: first a parametric state–space model is identified from data, and later on controllers are synthesized based on this model. Regarding the identification of a state–space model: given the data (U, X), we seek input and state matrices $(\widehat{B}, \widehat{A})$ so that they (approximately in the noisy case) satisfy the linear measurement Eq. (56). These can be obtained, e.g., as solution to the ordinary least squares problem

$$\begin{bmatrix} \widehat{B} & \widehat{A} \end{bmatrix} = \arg\min_{B,A} \left\| X_+ - \begin{bmatrix} B & A \end{bmatrix} \begin{bmatrix} U \\ X_- \end{bmatrix} \right\|_F = X_+ \begin{bmatrix} U \\ X_- \end{bmatrix}^\dagger, \tag{58}$$

where the solution is unique and given as above due to the identifiability condition (57).

Based on this identified pair of parameters $(\widehat{B}, \widehat{A})$ from (58) certainty equivalence controllers can be designed, that is, in the stabilization (54) and LQR (55) problems, the matrices (B, A) are replaced by their estimates $(\widehat{B}, \widehat{A})$. In case of noisy data, the uncertainty can be mitigated by robustifying the optimal control formulations. We refer to Dean et al. (2019), Mania et al. (2019), Treven et al. (2020) and Umenberger et al. (2019) for recent analysis, performance estimates for finite sample size, as well as various comparisons. Independently, this approach is also known as dynamic mode decomposition (DMD) in the nonlinear dynamics and fluids communities (Proctor et al., 2016).

In either the certainty-equivalent or the robust case, these approaches to data-driven control are indirect since they rely on an intermediate identification of a parametric state–space model. In what follows, we review direct approaches descending from the fundamental lemma and the subspace relations (56)–(57).

5.3.4. Optimal control parametrization by data matrices

The direct approach laid out by De Persis and Tesi (2019) uses the subspace relations (56)–(57) to parametrize the stabilization (54) and LQR (55) problems by raw data matrices. Namely, due to the rank condition (57), for any control gain matrix K, there is a (non-unique) $(L \times n)$ matrix G so that

$$\begin{bmatrix} K \\ I \end{bmatrix} = \begin{bmatrix} U \\ X_- \end{bmatrix} G, \tag{59}$$

and due to (56), the closed-loop matrix can be parameterized as

$$A + BK = \begin{bmatrix} B & A \end{bmatrix} \begin{bmatrix} K \\ I \end{bmatrix} \overset{(59)}{=} \begin{bmatrix} B & A \end{bmatrix} \begin{bmatrix} U \\ X_- \end{bmatrix} G \overset{(56)}{=} X_+ G . \tag{60}$$

This trick is as simple as it is ingenuous: it allows to replace the closed-loop matrix $A + BK$ by X_+G subject to the additional constraint (59), and the control gain can be recovered as $K = UG$. Thus, the stabilization (54) problem can be posed as

$$\underset{K,G,P \succ 0}{\text{find}} \quad (X_+G)P(X_+G)^\top - P \prec 0 \ \text{ and } \ \begin{bmatrix} K \\ I \end{bmatrix} = \begin{bmatrix} U \\ X_- \end{bmatrix} G. \tag{61}$$

Note that for $K = 0$, we obtain a data-driven stability test. After the change of variables $Q = GP$, condition (61) can be posed as a semidefinite constraint (De Persis & Tesi, 2019, Theorem 3)

$$\begin{bmatrix} X_-Q & X_+Q \\ \star & X_-Q \end{bmatrix} \succ 0 , \tag{62}$$

and, for any Q satisfying (62), $K = UQ(X_-Q)^{-1}$ is a stabilizing gain. The converse statement holds as well. Analogously, the LQR problem (55) can be parameterized by data raw matrices:

$$\begin{aligned} &\text{minimize} \quad \text{over } K, G, P \succeq I \quad \text{trace}\left(QP + K^\top RKP\right) \\ &\text{subject to} \quad (X_+G)P(X_+G)^\top - P + I \preceq 0 \\ &\qquad\qquad\quad \begin{bmatrix} K \\ I \end{bmatrix} = \begin{bmatrix} U \\ X_- \end{bmatrix} G \end{aligned} \tag{63}$$

After a substitution of variables, (63) can be solved as convex optimization problem (De Persis & Tesi, 2019, Theorem 4). This program (albeit convex) features matrix variables of size $L \times n$, i.e., dependent on the data size, which may be computationally challenging for a large data set. It is noteworthy that this approach also extends to the multivariable output feedback case by leveraging past outputs and inputs as states in a non-minimal realization (De Persis & Tesi, 2019, Section VI).

Note that this approach never constructs the underlying system matrices (B, A), as compared to identification (58). In fact, it uses only a data-based characterization of the closed-loop matrix $A + BK = X_+G$, and even here G solving (59) is not unique. This non-uniqueness provides an opportunity to further regularize the design in presence of noise. E.g., similar to (48), a least-norm solution minimizing $\|G\|_F$ can be sought by adding to (59) the orthogonality constraint (Dörfler, Tesi et al., 2021)

$$\left\| \left(I - \begin{bmatrix} U \\ X_- \end{bmatrix}^\dagger \begin{bmatrix} U \\ X_- \end{bmatrix} \right) G \right\| = 0 . \tag{64}$$

Thus, the resulting closed-loop matrix from (60) takes the form

$$\begin{bmatrix} B & A \end{bmatrix} \begin{bmatrix} K \\ I \end{bmatrix} = X^+G \overset{(64),\ (59)}{=} X_+ \begin{bmatrix} U \\ X_- \end{bmatrix}^\dagger \begin{bmatrix} K \\ I \end{bmatrix}.$$

Hence, the implicitly used state–space matrices are $\left[\begin{smallmatrix} B & A \end{smallmatrix}\right] = X_+ \left[\begin{smallmatrix} U \\ X_- \end{smallmatrix}\right]^\dagger$ and coincide with those in certainty-equivalence control and obtained from the least-squares estimate (58). Similar to Section 5.2.2, alternative regularizations for the direct data-driven control are conceivable (De Persis & Tesi, 2021b). We refer to Dörfler, Tesi et al. (2021) for an overview and comparison.

5.3.5. Data informativity

The innovative approach put forward by van Waarde et al. (2020) relies on data informativity: given the data (U, X), all linear systems that explain the data, i.e., that are compatible with the measurement Eq. (56), are parameterized by

$$\Sigma = \left\{ (B, A) | \ X_+ = \begin{bmatrix} B & A \end{bmatrix} \begin{bmatrix} U \\ X_- \end{bmatrix} \right\} . \tag{65}$$

The characterization of Σ is particularly interesting in the small-data limit when the rank condition (57) does not hold, and we cannot solve for a unique pair (B, A), i.e., Σ is not a singleton.

van Waarde et al. (2020) define the data (U, X) to be *informative for stabilization by state feedback* if there is a single feedback gain K so that $A + BK$ is Schur stable for all $(B, A) \in \Sigma$. If the rank condition (57) holds, then the data-informativity question may be approached as in Section 5.3.4. In absence of the rank condition (57), by studying the homogeneous solution set of the measurement Eq. (56), $\left\{ (B, A) | \ 0 = \left[B \ A \right] \left[\begin{smallmatrix} U \\ X \end{smallmatrix} \right] \right\}$, the authors are able to cast this problem as a semidefinitess condition (van Waarde et al., 2020, Theorem 17): namely, the data (U, X) are informative for stabilization by state feedback if and only if there is a matrix Q so that $X_- Q = X_- Q^\top$, (62) holds, and the stabilizing controller is obtained as $K = UQ(X_- Q)^{-1}$.

Hence, similar to De Persis and Tesi (2019) and van Waarde et al. (2020) arrive at the stabilization condition (62), though the derivation does not require the rank condition (57). In summary, the data informativity approach allows for stabilization even though the data is not sufficiently rich to identify the system. On the contrary, when studying data-informativity for the LQR problem (55), the rank condition (57) is required (van Waarde et al., 2020, Theorem 26), and a similar semidefinite program formulation as in (63) can be derived — though with optimization variables independent of the data size. van Waarde et al. (2020) also study the output feedback case and various system analysis questions from the view point of data informativity.

Applied to the problem of system identification, the informativity framework by van Waarde et al. (2020) leads to different identifiability definition and conditions than the ones presented in Section 3.2. The identifiability problem in van Waarde et al. (2020) concerns the special case of the input/state system (11), parameterized by the pair (A, B). Adapting the problem formulation to general input/output systems, parameterized by quadruple (A, B, C, D), requires the notion of equivalence classes in the space of parameters (A, B, C, D). A topic for future work is to extend the informativity framework to input/output systems and studying identifiability conditions.

5.3.6. Extensions beyond deterministic LTI systems

The articles by De Persis and Tesi (2019) and van Waarde et al. (2020) paved the way for manifold extensions to broader system classes, further analysis and design questions, as well as robustifications in case of inexact data. We will primarily review the latter here. In case the LTI dynamics (53) are subject to process disturbances w (in what follows referred to as "noise"),

$$\sigma x = Ax + Bu + w\,, \tag{66}$$

then the set of all systems explaining the data is characterized by all pairs of matrices (B, A) so that

$$X_+ = \begin{bmatrix} B & A \end{bmatrix} \begin{bmatrix} U \\ X_- \end{bmatrix} + W \tag{67}$$

for some realization $W = [w(0) \ w(1) \ \dots \ w(L-1)]$ of the unknown noise. In this case, by repeating the calculation (60), a data-driven parametrization of the closed-loop matrix is obtained as $A + BK = \left(X_+ - W\right) G$. From this point on, one may again pursue either a certainty-equivalence or a robust design.

De Persis and Tesi (2021b) pursue a certainty-equivalence approach, i.e., the control design is not robustified against noise. The authors provide quantitative bounds on the noise magnitude so that the certainty-equivalence design is stabilizing or a particular LQR suboptimality can be guaranteed. In a nutshell, the technical approach is as follows: in the semidefiniteness conditions (61) and (63), the matrix $X_+ G$ (encoding $A + BK$ in the noiseless case) has to be replaced by $\left(X_+ - W\right) G$ in the noisy case. The key question is then under which conditions feasibility of the semidefinite constraint without W (i.e., the certainty-equivalence case) implies feasibility with W. Since noise affects the semidefinite constraint through a product of the terms WGP, this approach also suggests to augment the LQR cost with a regularizer of the form $\mathrm{trace}(WGP(WG)^\top)$ to mitigate the effect of noise; see De Persis and Tesi (2021b, Section 5).

In the robust approach, the design seeks a stabilizing or optimal control for all W in a prescribed uncertainty set. A tractable and expressive uncertainty set is a quadratic matrix inequality proposed by Berberich, Koch, Scherer et al. (2020) and De Persis and Tesi (2019), where one assumes that all uncertainty realizations satisfy

$$\begin{bmatrix} I \\ W^\top \end{bmatrix}^\top \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \star & \Phi_{22} \end{bmatrix} \begin{bmatrix} I \\ W^\top \end{bmatrix} \succeq 0\,, \tag{68}$$

where $\Phi_{11} = \Phi_{11}^\top$ and $\Phi_{22} = \Phi_{22}^\top \prec 0$. For example, for $\Phi_{22} = -I$ and $\Phi_{12} = 0$ inequality (68) bounds the energy of W; or with $\mathbb{1}_{n \times n}$ being the $(n \times n)$ matrix of unit entries, $\Phi_{22} = -\frac{1}{n-1}(I - \frac{1}{n}\mathbb{1}_{n \times n})$, and $\Phi_{12} = 0$ (68) bounds the sample covariance of the noise (van Waarde, Camlibel et al., 2020). Different parameterizations of the uncertainty set (68) have been proposed; see van Waarde et al. (2021) for a discussion and conversions thereof. We also remark that Berberich, Scherer et al. (2020) and Bisoffi et al. (2021), Martin and Allgöwer (2021b) considered point-wise (in time) noise bounds to alleviate the potential conservatism of (68).

Given the data (U, X), a robustified version of the stabilization problem (54) can then be posed as finding a feedback gain K so that $A + BK$ is Schur stable for all (B, A) compatible with (67) and (68). At this point the data-driven robust design can be approached with established methods: De Persis and Tesi (2019) pose the problem as simultaneous satisfaction of a perturbed Lyapunov inequality with perturbation satisfying (68), and derive a robustly stabilizing controller for sufficiently large signal-to-noise ratio; Berberich, Koch, Scherer et al. (2020) recognize the problem setup as a linear fractional transformation, apply robust control methods, and later extend the approach to gray-box models including prior knowledge (Berberich, Scherer et al., 2020); alternatively, by inserting W from (67) into (68), van Waarde and Camlibel (2021) and van Waarde, Camlibel et al. (2020) arrive at a quadratic matrix inequality, pose the robust data-driven design as a simultaneous satisfaction of quadratic matrix inequalities, and solve it via a matrix-valued version of Finsler's and S-Lemma. Their conditions are non-conservative (necessary and sufficient) for the considered noise model and improve upon previous ones.

The above analysis has also been extended to stabilization of weakly nonlinear systems in the absolute stability setting, i.e., when system stabilization can be achieved by means of linear feedback and certified with a quadratic Lyapunov function. For example, to stabilize a Lur'e system $\sigma x = Ax + Bu + E\phi(x)$ with sector-bounded scalar nonlinearity $\phi(x)^\top(\phi(x) - x) \leq 0$, one can appeal to conceptually analogous methods as for a noisy system (67) subject to an ellipsoidal uncertainty set as in (68) (Luppi et al., 2021; van Waarde & Camlibel, 2021).

Also problems of (robust) invariance (Bisoffi et al., 2020a), stabilization of polynomial systems aided by sum-of-squares methods (Guo et al., 2020), or control design for bilinear (Bisoffi et al., 2020b), delayed (Rueda-Escobedo et al., 2020), switched (Rotulo et al., 2021), or rational (Strässer et al., 2020) dynamical systems lead to similar linear matrix inequalities.

5.3.7. Discussion

Many other works have followed up on these ideas resulting in a vibrant research arena. We will not provide an exhaustive overview. Rather we conclude with a few remarks.

First, many other instances of robust and optimal control can be formulated and parameterized as semidefinite optimization (respectively, feasibility) problems in terms of the closed-loop matrix $A + BK$ (Scherer & Weiland, 2000). Conceptually, all of these admit data-driven counterparts in case of exact data, and similar robustification methods as in Section 5.3.6 can be applied in case of noisy data. A possible caveat leading to computational challenges are the sizeable semidefinite programs.

Second, the above approaches model "noise" as a norm-bounded disturbance rather than as a stochastic process. A bridge between stochastic and worst-case noise models can be built by averaging data sets and constructing high-confidence norm bounds on W (De Persis & Tesi, 2021b, Section 6.2).

Third, the above approaches are all derived from the subspace relations (56)–(57) which again descend from the fundamental lemma. Though, the subsequent results are developed entirely in the state–space framework. Hence, most of the methods have been created under the dogma that a state–space representation is readily available with measurable states, and extensions to output feedback are often more conceptual than practically useful. However, in data-driven control design neither the state is available nor its dimension is a priori known, which provides a fruitful avenue for future research.

6. Concluding discussion and open problems

The behavioral system theory defines a system as a set of trajectories—the behavior—and is thus intrinsically amenable to data-driven approaches. Particular system representations, input/output partitioning of the variables, zero initial condition, and other assumptions are not imposed a priori. System properties and design problems are specified in terms of the behavior. Then, these properties can be checked, and analysis and design problems can be solved using data-driven methods.

The fundamental lemma (Lemma 2) and Corollary 5, reviewed in Section 3, give conditions for existence of a non-parametric data-driven representation of a linear time-invariant system. The condition of Corollary 5 is verifiable from the data and prior knowledge of the number of inputs, lag, and order of the system. It is a refinement of the fundamental lemma, which provides alternative sufficient conditions, assuming in addition a given input/output partitioning of the variables and controllability of the system.

The data-driven representation allows approaching system theory, signal processing, and control problems using basic linear algebra. It leads to general, simple, and practical solution methods. This was illustrated in the paper by applying it on a problem of data-driven missing data estimation. The resulting method assumes only linear time-invariant system dynamics and has no hyperparameters. In case of noisy data, generated in the errors-in-variables setup, the maximum-likelihood estimator is obtained by a Hankel structured low-rank approximation and completion problem. The maximum-likelihood estimation problem is nonconvex, however, ℓ_1-norm regularization provides an effective convex relaxation.

The fundamental lemma has long served as a cornerstone of indirect data-driven control, that is, sequential system identification and control. Recently, multiple direct data-driven control formulations have sparked from the fundamental lemma and the associated non-parametric system representation. These can be loosely classified as implicit and explicit approaches, as represented by the DeePC and data-driven LQR approaches. The approaches are equally amenable to certainty-equivalence and robust control implementations. Within the vast realm of data-driven control, the approaches based on the fundamental lemma are remarkable, as they are amenable to theoretic analysis and certification, but they are also computationally tractable, require only few data samples, and robustly implementable in real-time and safety-critical physical control systems.

While the initial fundamental lemma dates back almost 20 years, it has recently given rise to a blossoming literature in the vibrant research arena of data-driven control. On top of the manifold open problems already pointed out earlier, we see the following promising and important avenues for future research.

On the theory side, the presented data-driven approaches are based on an inherent LTI model specification—the non-parametric model representation—and by means of robustifying and adapting the optimization methods they have been successfully applied to stochastic, nonlinear, and time-varying systems. Though what is missing is a bottom-up approach extending behavioral systems and the non-parametric representation to the stochastic and nonlinear domain. Furthermore, most of the presented approaches rely on sequential exploration (data collection) and control (exploitation). Though the overall goal should be direct, online, and adaptive data-driven approaches relying on partial and noisy output measurements.

On the computational side, data-driven methods are currently based on batch computation without exploitation of the special structure of the data matrices. Because of this, the computational complexity of the data-driven methods does not compare favorably with the one of model-based methods. A topic for further research is therefore development of efficient computational methods as well as recursive methods that are suitable for online computation. Other important topics are sensitivity analysis of the algorithms in case of noisy data, automatic tuning of the hyperparameters, and selecting the matrix structure. In particular, we presented three different matrix structures: (mosaic) Hankel, Page, and trajectory. Preliminary empirical evidence suggests that they are suitable in different types of problems: the Hankel matrix in model identification problems while the Page and trajectory matrices in data-driven control via DeePC. More experiments as well as theoretical analysis are needed in order to find definitive guidelines for which matrix structure to use under which assumptions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors wish to thank Jeremy Coulson, Henk van Waarde, Julian Berberich, Andrea Martin, Eduardo Prieto, Claudio de Persis, Pietro Tesi, Paolo Rapisarda, Rodolphe Sepulchre, and John Lygeros for many discussions leading up to and improving this survey paper.

The research leading to these results has received funding from the European Research Council (ERC) Grant agreement number 258581 "Structured low-rank approximation: Theory, algorithms, and applications"; the Fond for Scientific Research Vlaanderen (FWO) projects G028015N "Decoupling multivariate polynomials in nonlinear system identification" and G090117N "Block-oriented nonlinear identification using Volterra series"; and the Fonds de la Recherche Scientifique (FNRS), Belgium – FWO under Excellence of Science (EOS) Project no 30468160 "Structured low-rank matrix/tensor approximation: numerical optimization-based algorithms and applications".

Appendix A. Proof of the dimension formula (1)

Let $\mathscr{B}_{\text{ss}}(A, B, C, D)$ be a minimal input/state/output representation of the system $\mathscr{B}$ (*cf.* Section 2.3). For any $w \in \mathscr{B}|_L$, there is $x(1) = x_{\text{ini}} \in \mathbb{R}^n$, such that

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

This system of equations can be written more compactly as

$$w = \Pi_L \begin{bmatrix} 0 & I \\ \mathscr{O}_L & \mathscr{C}_L \end{bmatrix} \begin{bmatrix} x(1) \\ u \end{bmatrix} =: M_L \begin{bmatrix} x(1) \\ u \end{bmatrix}, \tag{A.1}$$

where $\Pi_L \in \mathbb{R}^{qL\times qL}$ is a permutation matrix (determined by Π and the re-grouping of the variables in the left-hand-side and right-hand-side of (A.1)), $\mathscr{O}_L \in \mathbb{R}^{pL\times n}$ is the extended observability matrix with L block rows, defined in (6), and $\mathscr{C}_L \in \mathbb{R}^{pL\times mL}$ is the *convolution matrix* with L block rows

$$\mathscr{C}_L := \begin{bmatrix} h(0) & 0 & \cdots & 0 \\ h(1) & h(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ h(L-1) & \cdots & h(1) & h(0) \end{bmatrix}, \tag{A.2}$$

constructed from the Markov parameters

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

of the system. From (A.1), it follows that

$$\dim \mathscr{B}|_L = \operatorname{rank} M_L.$$

Since the representation is minimal and $L \geq \mathbf{l}(\mathscr{B})$, the extended observability matrix $\mathscr{O}_L$ is full column rank n. Then, due to the lower-triangular block-structure of M_L and the identity block, M_L is also full column rank. Therefore,

$$\dim \mathscr{B}|_L = \operatorname{rank} M_L = mL + n.$$

Appendix B. Proof of Lemma 1

We provide a state–space and a representation-free proof.

B.1. Using an input/state/output representation

Let $\mathscr{B}_{\text{ss}}(A, B, C, D)$ be a minimal input/state/output representation of the system $\mathscr{B}$. Since w_{ini} is a trajectory of $\mathscr{B}|_{T_{\text{ini}}}$, there is an $x(1) = x_{\text{ini}} \in \mathbb{R}^n$, such that

$$y_{\text{ini}} = \mathscr{O}_{T_{\text{ini}}} x_{\text{ini}} + \mathscr{C}_{T_{\text{ini}}} u_{\text{ini}}, \tag{B.1}$$

where $\mathscr{O}_{T_{\text{ini}}}$ is defined in (6) and $\mathscr{C}_{T_{\text{ini}}}$ is defined in (A.2). Moreover, the assumption that $\mathscr{B}(A, B, C, D)$ is a minimal representation and $T_{\text{ini}} \geq \mathbf{l}(\mathscr{B})$ imply that the extended observability matrix $\mathscr{O}_{T_{\text{ini}}}$ has full column rank. Therefore, the system of equations (B.1) has a unique solution x_{ini}. The initial condition $x(T_{\text{ini}} + 1)$ for the trajectory w_{f} is uniquely determined by x_{ini} and u_{ini}

$$x(T_{\text{ini}} + 1) = A^{T_{\text{ini}}} x_{\text{ini}} + \begin{bmatrix} A^{T_{\text{ini}}-1}B & A^{T_{\text{ini}}-2}B & \cdots & AB & B \end{bmatrix} u_{\text{ini}}.$$

The uniqueness of y_{f} follows from the uniqueness of $x(T_{\text{ini}} + 1)$.

B.2. A representation-free proof

Let $n := \mathbf{n}(\mathscr{B})$. By the dimension formula (1), there is a full column rank matrix $B \in \mathbb{R}^{q(T_{\text{ini}}+L)\times(m(T_{\text{ini}}+L)+n)}$, such that

$$w_{\text{ini}} \wedge (u, y) = Bg,$$

for some $g \in \mathbb{R}^{m(T_{\text{ini}}+L)+n}$, *i.e.*, the columns of B form a basis for $\mathscr{B}|_{T_{\text{ini}}+L}$. Denote with $B' \in \mathbb{R}^{(qT_{\text{ini}}+mL)\times(m(T_{\text{ini}}+L)+n)}$ the submatrix of B obtained by selecting the rows of B corresponding to w_{ini} and u. The simulation problem (8) has a unique solution if and only if the system of equations

$$\begin{bmatrix} w_{\text{ini}} \\ u \end{bmatrix} = B'g \tag{B.2}$$

has a unique solution g. A necessary and sufficient condition for uniqueness of a solution of (B.2) is that B' is full column rank. By the assumption $T_{\text{ini}} \geq \mathbf{l}(\mathscr{B})$, using (1) and the fact that u is a free variable, we have

$$\begin{aligned} \operatorname{rank} B' &= \dim \left\{ \left[\begin{smallmatrix} w_{\text{ini}} \\ u \end{smallmatrix}\right] \mid w_{\text{ini}} \in \mathscr{B}|_{T_{\text{ini}}} \text{ and } u \in \mathbb{R}^{mL} \right\} \\ &= mT_{\text{ini}} + mL + n = \operatorname{col\,dim} B', \end{aligned}$$

so that, B' is indeed full column rank.

Appendix C. Proofs of Corollary 5

The fact that $\operatorname{image} \mathscr{H}_L(\mathscr{W}_{\text{d}}) \subseteq \mathscr{B}|_L$ follows from the linear time-invariance property of $\mathscr{B}$ and the exactness of the data $\mathscr{W}_{\text{d}}$. In order to prove that equality holds if and only if the generalized persistency of excitation condition (14) holds, note that

$$\dim \operatorname{image} \mathscr{H}_L(\mathscr{W}_{\text{d}}) = \operatorname{rank} \mathscr{H}_L(\mathscr{W}_{\text{d}}) \tag{C.1}$$

and (1). Since, $\operatorname{image} \mathscr{H}_L(\mathscr{W}_{\text{d}})$ is included in $\mathscr{B}|_L$, they are equal if and only if their dimensions are equal. The result follows from (C.1) and (1).

References

Agarwal, A., Amjad, M. J., Shah, D., & Shen, D. (2018). Model agnostic time series analysis via matrix estimation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, *2*(3), 1–39.

Alexandru, A. B., Tsiamis, A., & Pappas, G. J. (2021). Encrypted distributed lasso for sparse data predictive control. ArXiv preprint arXiv:2104.11632.

Allibhoy, A., & Cortés, J. (2020). Data-based receding horizon control of linear network systems. *IEEE Control Systems Letters*, *5*(4), 1207–1212.

Alpago, D., Dörfler, F., & Lygeros, J. (2020). An extended Kalman filter for data-enabled predictive control. *IEEE Control Systems Letters*, *4*(4), 994–999.

Alsalti, M., Berberich, J., Lopez, V. G., Allgöwer, F., & Müller, M. A. (2021). Data-based system analysis and control of flat nonlinear systems. ArXiv preprint arXiv:2103.02892.

Anderson, J., Doyle, J. C., Low, S. H., & Matni, N. (2019). System level synthesis. *Annual Reviews in Control*, *47*, 364–393.

Anderson, B. D., & Moore, J. B. (2007). *Optimal control: Linear quadratic methods*. Courier Corporation.

Antsaklis, P. J., & Michel, A. (1997). *Linear systems*. McGraw-Hill.

Baggio, G., Bassett, D. S., & Pasqualetti, F. (2021). Data-driven control of complex networks. *Nature Communications*, *12*(1), 1–13.

Baggio, G., Katewa, V., & Pasqualetti, F. (2019). Data-driven minimum-energy controls for linear systems. *IEEE Control Systems Letters*, *3*(3), 589–594.

Baggio, G., & Pasqualetti, F. (2020). Learning minimum-energy controls from heterogeneous data. In *2020 American control conference* (pp. 3991–3996). IEEE.

Baros, S., Chang, C.-Y., Colon-Reyes, G. E., & Bernstein, A. (2020). Online data-enabled predictive control. ArXiv preprint arXiv:2003.03866.

Berberich, J., & Allgöwer, F. (2020). A trajectory-based framework for data-driven system analysis and control. In *European control conf.* (pp. 1365–1370).

Berberich, J., Koch, A., Scherer, C. W., & Allgöwer, F. (2020). Robust data-driven state-feedback design. In *2020 American control conference* (pp. 1532–1538). IEEE.

Berberich, J., Köhler, J., Müller, M. A., & Allgöwer, F. (2020a). Data-driven tracking MPC for changing setpoints. *IFAC-PapersOnLine*, *53*(2), 6923–6930.

Berberich, J., Köhler, J., Müller, M. A., & Allgöwer, F. (2020b). Robust constraint satisfaction in data-driven MPC. In *2020 59th IEEE conference on decision and control* (pp. 1260–1267).

Berberich, J., Köhler, J., Müller, M. A., & Allgöwer, F. (2021a). Data-driven model predictive control: closed-loop guarantees and experimental results. *At-Automatisierungstechnik*, *69*(7), 608–618.

Berberich, J., Köhler, J., Müller, M. A., & Allgöwer, F. (2021b). Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, *66*(4), 1702–1717.

Berberich, J., Köhler, J., Müller, M. A., & Allgöwer, F. (2021c). Linear tracking MPC for nonlinear systems parts I & II. ArXiv preprints arXiv:2105.08560 and arXiv:2105.08567.

Berberich, J., Köhler, J., Müller, M. A., & Allgöwer, F. (2021d). On the design of terminal ingredients for data-driven MPC. ArXiv preprint arXiv:2101.05573.

Berberich, J., Scherer, C. W., & Allgöwer, F. (2020). Combining prior knowledge and data for robust controller design. ArXiv preprint arXiv:2009.05253.

Bertsimas, D., & Copenhaver, M. S. (2018). Characterization of the equivalence of robustification and regularization in linear and matrix regression. *European Journal of Operational Research*, *270*(3), 931–942.

Bianchin, G., Vaquero, M., Cortes, J., & Dall'Anese, E. (2021). Data-driven synthesis of optimization-based controllers for regulation of unknown linear systems. ArXiv preprint arXiv:2103.16067.

Bisoffi, A., De Persis, C., & Tesi, P. (2020a). Controller design for robust invariance from noisy data. ArXiv preprint arXiv:2007.13181.

Bisoffi, A., De Persis, C., & Tesi, P. (2020b). Data-based stabilization of unknown bilinear systems with guaranteed basin of attraction. *Systems & Control Letters*, *145*, Article 104788.

Bisoffi, A., De Persis, C., & Tesi, P. (2021). Trade-offs in learning controllers from noisy data. ArXiv preprint arXiv:2103.08629.

Bongard, J., Berberich, J., Köhler, J., & Allgöwer, F. (2021). Robust stability analysis of a simple data-driven model predictive control approach. ArXiv preprint arXiv:2103.00851.

Borrelli, F., Bemporad, A., & Morari, M. (2017). *Predictive control for linear and hybrid systems*. Cambridge University Press.

Box, G., & Jenkins, G. (1976). *Time series analysis: Forecasting and control*. Holden-Day.

Brunton, S., Proctor, J., & Kutz, N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, *113*, 3932–3937.

Carlet, P., A., F., Bolognani, S., & Dörfler, F. (2021). Data-driven continuous-set predictive current control for synchronous motor drives. *IEEE Transactions on Power Electronics*, submitted for publication.

Carlet, P. G., Favato, A., Bolognani, S., & Dörfler, F. (2020). Data-driven predictive current control for synchronous motor drives. In *2020 IEEE energy conversion congress and exposition* (pp. 5148–5154).

Chiuso, A., & Pillonetto, G. (2019). System identification: A machine learning perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, *2*, 281–304.

Coulson, J., Lygeros, J., & Dörfler, F. (2019a). Data-enabled predictive control: In the shallows of the deepc. In *European control conference* (pp. 307–312).

Coulson, J., Lygeros, J., & Dörfler, F. (2019b). Regularized and distributionally robust data-enabled predictive control. In *Proc. of IEEE conf. on decision and control* (pp. 7165–7170).

Coulson, J., Lygeros, J., & Dörfler, F. (2020). Distributionally robust chance constrained data-enabled predictive control. in press. Available at https://arxiv.org/abs/2006.01702,

Cutler, C. R., & Ramaker, B. L. (1980). Dynamic matrix control – A computer control algorithm. In *Joint automatic control conference* (p. 72).

Damen, A., Van den Hof, P., & Hajdasinski, A. (1982). Approximate realization based upon an alternative to the Hankel matrix: the Page matrix. *Control Systems Letters*, *2*, 202–208.

De Persis, C., & Tesi, P. (2019). Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, *65*(3), 909–924.

De Persis, C., & Tesi, P. (2021a). Designing experiments for data-driven control of nonlinear systems. ArXiv preprint arXiv:2103.16509.

De Persis, C., & Tesi, P. (2021b). Low-complexity learning of linear quadratic regulators from noisy data. *Automatica*, *128*, Article 109548.

Dean, S., Mania, H., Matni, N., Recht, B., & Tu, S. (2019). On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 1–47.

Dörfler, F., Coulson, J., & Markovsky, I. (2021a). Bridging direct & indirect data-driven control formulations via regularizations and relaxations. https://arxiv.org/abs/2101.01273.

Dörfler, F., Coulson, J., & Markovsky, I. (2021b). *Bridging direct & indirect data-driven control formulations via regularizations and relaxations*: Technical Report, arXiv:2101.01273. URL: https://arxiv.org/abs/2101.01273.

Dörfler, F., Tesi, P., & De Persis, C. (2021). On the certainty-equivalence approach to direct data-driven LQR design. ArXiv preprint arXiv:2109.06643.

Dreesen, P., & Markovsky, I. (2019). Data-driven simulation using the nuclear norm heuristic. In *In proceedings of the international conference on acoustics, speech, and signal processing*. Brighton, UK.

El Ghaoui, L., & Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, *18*(4), 1035–1064.

Elokda, E., Coulson, J., Beuchat, P., Lygeros, J., & Dörfler, F. (2019). Data-enabled predictive control for quadcopters. *International Journal of Robust and Nonlinear Control*, http://dx.doi.org/10.1002/rnc.5686, in press.

Fabiani, F., & Goulart, P. J. (2020). The optimal transport paradigm enables data compression in data-driven robust control. ArXiv preprint arXiv:2005.09393.

Favoreel, W., De Moor, B., & Gevers, M. (1999). SPC: subspace predictive control. *IFAC Proceedings Volumes*, *32*(2), 4004–4009.

Fazel, M. (2002). *Matrix rank minimization with applications*. (Ph.D. thesis), Stanford University.

Feron, E., Balakrishnan, V., Boyd, S., & El Ghaoui, L. (1992). Numerical methods for H_2 related problems. In *1992 American control conference* (pp. 2921–2922). IEEE.

Fiedler, F., & Lucia, S. (2021). On the relationship between data-enabled predictive control and subspace predictive control. In *European control conference*.

Frazzoli, E., Dahleh, M. A., & Feron, E. (2005). Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Transactions on Robotics*, *21*(6), 1077–1091.

Fujisaki, Y., Duan, Y., & Ikeda, M. (2004). System representation and optimal control in input-output data space. *IFAC Proceedings Volumes*, *37*(11), 185–190.

Furieri, L., Guo, B., Martin, A., & Ferrari-Trecate, G. (2021a). A behavioral input-output parametrization of control policies with suboptimality guarantees. ArXiv preprint arXiv:2102.13338.

Furieri, L., Guo, B., Martin, A., & Ferrari-Trecate, G. (2021b). Near-optimal design of safe output feedback controllers from noisy data. ArXiv preprint arXiv:2105.10280.

Furieri, L., Zheng, Y., Papachristodoulou, A., & Kamgarpour, M. (2019). An input–output parametrization of stabilizing controllers: Amidst youla and system level synthesis. *IEEE Control Systems Letters*, *3*(4), 1014–1019.

Garcia, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: Theory and practice – A survey. *Automatica*, *25*(3), 335–348.

Golub, G., & Van Loan, C. (1996). *Matrix computations* (3rd ed.). Johns Hopkins University Press.

Grant, M., & Boyd, S. (2008). CVX: Matlab software for disciplined convex programming. URL: stanford.edu/∼boyd/cvx.

Gray, A., Gao, Y., Lin, T., Hedrick, J. K., Tseng, H. E., & Borrelli, F. (2012). Predictive control for agile semi-autonomous ground vehicles using motion primitives. In *2012 American control conference* (pp. 4239–4244). IEEE.

Guo, M., De Persis, C., & Tesi, P. (2020). Data-driven stabilization of nonlinear polynomial systems with noisy data. ArXiv preprint arXiv:2011.07833.

Hauswirth, A., Bolognani, S., Hug, G., & Dörfler, F. (2021). Optimization algorithms as robust feedback controllers. URL: http://arxiv.org/abs/2103.11329. submitted for publication. Available at http://arxiv.org/abs/2103.11329.

Heinig, G. (1995). Generalized inverses of Hankel and Toeplitz mosaic matrices. *Linear Algebra and its Applications*, *216*, 43–59.

Hewing, L., Wabersich, K., Menner, M., & Zeilinger, M. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*, 269–296.

Hjalmarsson, H. (2005). From experiment design to closed-loop control. *Automatica*, *41*(3), 393–438.

Hou, Z.-S., & Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, *235*, 3–35.

Huang, L., Coulson, J., Lygeros, J., & Dörfler, F. (2019). Data-enabled predictive control for grid-connected power converters. In *2019 IEEE 58th conference on decision and control (CDC)* (pp. 8130–8135). IEEE.

Huang, L., Coulson, J., Lygeros, J., & Dörfler, F. (2021). Decentralized data-enabled predictive control for power system oscillation damping. *IEEE Transactions on Control Systems Technology*, in press. Available at https://arxiv.org/abs/1911.12151.

Huang, L., Jianzhe, Z., Lygeros, J., & Dörfler, F. (2021). Robust data-enabled predictive control: Tractable formulations and performance guarantees. ArXiv preprint arXiv:2105.07199.

Huang, B., & Kadali, R. (2008). *Dynamic modeling, predictive control and performance monitoring: A data-driven subspace approach*. Springer.

Huang, L., Zhen, J., Lygeros, J., & Dörfler, F. (2021). Quadratic regularization of data-enabled predictive control: Theory and application to power converter experiments. In *IFAC symposium on system identification*. in press. Available at https://arxiv.org/abs/2012.04434.

Iannelli, A., Yin, M., & Smith, R. S. (2020). Experiment design for impulse response identification with signal matrix models. ArXiv preprint arXiv:2012.08126.

Ikeda, M., Fujisaki, Y., & Hayashi, N. (2001). A model-less algorithm for tracking control based on input-output data. *Nonlinear Analysis. Theory, Methods & Applications*, *47*(3), 1953–1960.

Jud, D., Kerscher, S., Wermelinger, M., Jelavic, E., Egli, P., Leemann, P., Hottiger, G., & Hutter, M. (2021). Heap-the autonomous walking excavator. *Automation in Construction*, *129*, Article 103783.

Kailath, T., Sayed, A. H., & Hassibi, B. (2000). *Linear estimation*. Prentice Hall.

Kaiser, E., Kutz, J. N., & Brunton, S. L. (2018). Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society of London, Series A (Mathematical and Physical Sciences)*, *474*(2219), Article 20180335.

Koch, A., Berberich, J., Köhler, J., & Allgöwer, F. (2020). Determining optimal input-output properties: A data-driven approach. arXiv:arXiv:2002.03882.

Krishnan, V., & Pasqualetti, F. (2021). On direct vs indirect data-driven predictive control. ArXiv preprint arXiv:2103.14936.

Kuhn, D., Esfahani, P., Nguyen, V., & Shafieezadeh-Abadeh, S. (2019). Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations research & management science in the age of analytics* (pp. 130–166). INFORMS.

Landau, I., Rey, D., Karimi, A., Voda, A., & Franco, A. (1995). A flexible transmission system as a benchmark for robust digital control. *European Journal of Control*, *1*(2), 77–96.

Lian, Y., & Jones, C. (2021a). Nonlinear data-enabled prediction and control. arXiv:2101.03187.

Lian, Y., & Jones, C. N. (2021b). From system level synthesis to robust closed-loop data-enabled predictive control. ArXiv preprint arXiv:2102.06553.

Lian, Y., Shi, J., Koch, M. P., & Jones, C. N. (2021). Adaptive robust data-driven building control via bi-level reformulation: an experimental result. ArXiv preprint arXiv:2106.05740.

Lian, Y., Wang, R., & Jones, C. (2021). Koopman based data-driven predictive control. arXiv:2102.05122.

Lu, X., Chen, H., Gao, B., Zhang, Z., & Jin, W. (2014). Data-driven predictive gearshift control for dual-clutch transmissions and FPGA implementation. *IEEE Transactions on Industrial Electronics*, *62*(1), 599–610.

Lundström, P., Lee, J. H., Morari, M., & Skogestad, S. (1995). Limitations of dynamic matrix control. *Computers & Chemical Engineering*, *19*(4), 409–421.

Luppi, A., De Persis, C., & Tesi, P. (2021). On data-driven stabilization of systems with quadratic nonlinearities. ArXiv preprint arXiv:2103.15631.

Mania, H., Tu, S., & Recht, B. (2019). Certainty equivalence is efficient for linear quadratic control. ArXiv preprint arXiv:1902.07826.

Markovsky, I. (2008). Structured low-rank approximation and its applications. *Automatica*, *44*(4), 891–909.

Markovsky, I. (2012). How effective is the nuclear norm heuristic in solving data approximation problems? In *Proc. of the 16th IFAC symposium on system identification* (pp. 316–321). Brussels.

Markovsky, I. (2013). A software package for system identification in the behavioral setting. *Control Engineering Practice*, *21*, 1422–1436.

Markovsky, I. (2014). Recent progress on variable projection methods for structured low-rank approximation. *Signal Processing*, *96PB*, 406–419.

Markovsky, I. (2015). An application of system identification in metrology. *Control Engineering Practice*, *43*, 85–93.

Markovsky, I. (2017). A missing data approach to data-driven filtering and control. *IEEE Transactions on Automatic Control*, *62*, 1972–1978.

Markovsky, I. (2019). *Low-rank approximation: Algorithms, implementation, applications* (2nd ed.). Springer.

Markovsky, I., & De Moor, B. (2005). Linear dynamic filtering with noisy input and output. *Automatica*, *41*(1), 167–171.

Markovsky, I., & Dörfler, F. (2020). *Identifiability in the behavioral setting*. Vrije Universiteit Brussel, URL: http://homepages.vub.ac.be/~imarkovs/publications/identifiability.pdf.

Markovsky, I., & Dörfler, F. (2021). Data-driven dynamic interpolation and approximation. *Automatica*.

Markovsky, I., & Rapisarda, P. (2008). Data-driven simulation and control. *International Journal of Control*, *81*(12), 1946–1959.

Markovsky, I., & Usevich, K. (2013). Structured low-rank approximation with missing data. *SIAM Journal of Mathematical Analysis*, *34*(2), 814–830.

Markovsky, I., Willems, J. C., Rapisarda, P., & De Moor, B. (2005). Algorithms for deterministic balanced subspace identification. *Automatica*, *41*(5), 755–766.

Markovsky, I., Willems, J. C., Van Huffel, S., & De Moor, B. (2006). *Exact and approximate modeling of linear systems: A behavioral approach*. SIAM.

Marro, G., Prattichizzo, D., & Zattoni, E. (2002). Geometric insight into discrete-time cheap and singular linear quadratic riccati (LQR) problems. *IEEE Transactions on Automatic Control*, *47*(1), 102–107.

Martin, T., & Allgöwer, F. (2021a). Data-driven inference on optimal input-output properties of polynomial systems with focus on nonlinearity measures. arXiv:arXiv:2103.10306.

Martin, T., & Allgöwer, F. (2021b). Dissipativity verification with guarantees for polynomial systems from noisy input-state data. *IEEE Control Systems Letters*, *5*(4), 1399–1404. http://dx.doi.org/10.1109/LCSYS.2020.3037842.

Maupong, T., Mayo-Maldonado, J. C., & Rapisarda, P. (2017). On Lyapunov functions and data-driven dissipativity. *IFAC-PapersOnLine*, *50*(1), 7783–7788.

Mishra, V., Markovsky, I., & Grossmann, B. (2020). Data-driven tests for controllability. *Control Systems Letters*, *5*, 517–522.

Monshizadeh, N. (2020). Amidst data-driven model reduction and control. *IEEE Control Systems Letters*, *4*(4), 833–838.

Nonhoff, M., & Müller, M. A. (2021). Data-driven online convex optimization for control of dynamical systems. ArXiv preprint arXiv:2103.09127.

Parikh, N., & Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, *1*, 123–231.

Pillai, H., & Willems, J. (1999). The behavioural approach to distributed systems. In *Proc. 38th IEEE conference on decision and control, Vol. 1* (pp. 626–630).

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., & Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, *50*(3), 657–682.

Polderman, J., & Willems, J. C. (1998). *Introduction to mathematical systems theory*. Springer-Verlag.

Proctor, J. L., Brunton, S. L., & Kutz, J. N. (2016). Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, *15*(1), 142–161.

Rapisarda, P., & Willems, J. C. (1997). State maps for linear systems. *SIAM Journal on Control and Optimization*, *35*(3), 1053–1091.

Rawlings, J. B., Mayne, D. Q., & Diehl, M. (2017). *Model predictive control: theory, computation, and design, Vol. 2*. Nob Hill Publishing Madison, WI.

Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, *2*, 253–279.

Romer, A., Berberich, J., Köhler, J., & Allgöwer, F. (2019). One-shot verification of dissipativity properties from input–output data. *IEEE Control Systems Letters*, *3*(3), 709–714.

Roorda, B., & Heij, C. (1995). Global total least squares modeling of multivariate time series. *IEEE Transactions on Automatic Control*, *40*(1), 50–63.

Rosa, T. E., & Jayawardhana, B. (2021). On the one-shot data-driven verification of dissipativity of lti systems with general quadratic supply rate function. ArXiv preprint arXiv:2104.03108.

Rotulo, M., De Persis, C., & Tesi, P. (2021). Online learning of data-driven controllers for unknown switched linear systems. ArXiv preprint arXiv:2105.11523.

Rueda-Escobedo, J. G., Fridman, E., & Schiffer, J. (2020). Data-driven control for linear discrete-time delay systems. ArXiv preprint arXiv:2010.02657.

Rueda-Escobedo, J., & Schiffer, J. (2020). Data-driven internal model control of second-order discrete Volterra systems. arXiv:2003.14158.

Salvador, J. R., Ramírez, D. R., Alamo, T., de la Peña, D. M. n., & Garcia-Marin, G. (2019). Data driven control: an offset free approach. In *2019 18th European control conference* (pp. 23–28). IEEE.

Scherer, C., & Weiland, S. (2000). Linear matrix inequalities in control. *Lecture Notes, Dutch Institute for Systems and Control, Delft, the Netherlands*, *3*(2).

Schwarz, J., Micheli, F., Hudoba de Badyn, M., & Smith, R. (2019). *Data-driven control of buildings and energy hubs* (Semester Thesis), ETH Zurich, Available at https://www.research-collection.ethz.ch/.

Strässer, R., Berberich, J., & Allgöwer, F. (2020). Data-driven control of nonlinear systems: Beyond polynomial dynamics. ArXiv preprint arXiv:2011.11355.

Tischhauser, F., Egli, P., Coulson, J., Hutter, M., & Dörfler, F. (2019). *Data-enabled predictive control of robotic systems* (Semester Thesis), ETH Zurich, Available at https://www.research-collection.ethz.ch/.

Treven, L., Curi, S., Mutny, M., & Krause, A. (2020). Learning controllers for unstable linear quadratic regulators from a single trajectory. ArXiv preprint arXiv:2006.11022.

Umenberger, J., Ferizbegovic, M., Schön, T., & Hjalmarsson, H. (2019). Robust exploration in linear quadratic reinforcement learning. In *33rd annual conference on neural information processing systems*.

Usevich, K., & Markovsky, I. (2014). Variable projection for affinely structured low-rank approximation in weighted 2-norms. *Journal of Computational and Applied Mathematics*, *272*, 430–448.

Vajpayee, V., Mukhopadhyay, S., & Tiwari, A. P. (2017). Data-driven subspace predictive control of a nuclear reactor. *IEEE Transactions on Nuclear Science*, *65*(2), 666–679.

Van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems: Theory, implementation, applications*. Boston: Kluwer.

van Waarde, H., Camlibel, K., & Mesbahi, M. (2020). From noisy data to feedback controllers: Non-conservative design via a matrix S-lemma. arXiv:2006.00870.

van Waarde, H. J., De Persis, C., Camlibel, M. K., & Tesi, P. (2020). Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Systems Letters*, *4*, 602–607.

Verhaegen, M., & Dewilde, P. (1992). Subspace model identification, part 2: Analysis of the output-error state-space model identification algorithm. *International Journal of Control*, *56*, 1187–1210.

Verhoek, C., Tóth, R., Haesaert, S., & Koch, A. (2021). Fundamental lemma for data-driven analysis of linear parameter-varying systems. ArXiv preprint arXiv:2103.16171.

van Waarde, H. J. (2021). Beyond persistency of excitation: Online experiment design for data-driven modeling and control. ArXiv preprint arXiv:2102.11193.

van Waarde, H. J., & Camlibel, M. K. (2021). A Matrix Finsler's lemma with applications to data-driven control. ArXiv preprint arXiv:2103.13461.

van Waarde, H. J., Camlibel, M. K., Rapisarda, P., & Trentelman, H. L. (2021). Data-driven dissipativity analysis: application of the matrix S-lemma. *Control Systems Magazine*, submitted for publication.

van Waarde, H. J., Eising, J., Trentelman, H. L., & Camlibel, M. K. (2020). Data informativity: a new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, *65*(11), 4753–4768.

Wegner, F., Coulson, J., Hudoba de Badyn, M., Lygeros, J., & Trimpe, S. (2021). *Data-enabled predictive control of a 12t excavator* (Master Thesis), ETH Zurich, Available at https://www.research-collection.ethz.ch/.

Willems, J. C. (1986). From time series to linear system—Part I. Finite dimensional linear time invariant systems. *Automatica*, *22*(5), 561–580.

Willems, J. C. (1986, 1987). From time series to linear system—Part I. Finite dimensional linear time invariant systems, part II. Exact modelling, part III. approximate modelling. *Automatica*, *22, 23*, 561–580, 675–694, 87–115.

Willems, J. C. (1991). Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control*, *36*(3), 259–294.

Willems, J. C. (2007a). The behavioral approach to open and interconnected systems: Modeling by tearing, zooming, and linking. *Control Systems Magazine*, *27*, 46–99.

Willems, J. C. (2007b). In control, almost from the beginning until the day after tomorrow. *European Journal of Control*, *13*, 71–81.

Willems, J. C., Rapisarda, P., Markovsky, I., & De Moor, B. (2005). A note on persistency of excitation. *Control Letters*, *54*(4), 325–329.

Xu, H., Caramanis, C., & Mannor, S. (2010). Robust regression and lasso. *IEEE Transactions on Information Theory*, *56*(7), 3561–3574.

Xu, L., Turan, M. S., Guo, B., & Ferrari-Trecate, G. (2021). A data-driven convex programming approach to worst-case robust tracking controller design. ArXiv preprint arXiv:2102.11918.

Xue, A., & Matni, N. (2020). Data-driven system level synthesis. arXiv:2011.10674.

Yang, H., & Li, S. (2013). A new method of direct data-driven predictive controller design. In *2013 9th Asian control conference (ASCC)* (pp. 1–6). IEEE.

Yin, M., Iannelli, A., & Smith, R. (2020a). Maximum likelihood estimation in data-driven modeling and control.

Yin, M., Iannelli, A., & Smith, R. S. (2020b). Maximum likelihood signal matrix model for data-driven predictive control. ArXiv preprint arXiv:2012.04678.

Yu, Y., Talebi, S., van Waarde, H., Topcu, U., Mesbahi, M., & Açıkmeşe, B. (2021). On controllability and persistency of excitation in data-driven control: Extensions of Willems' fundamental lemma. arXiv:2102.02953v1.

Zeng, J.-s., Gao, C.-h., & Su, H.-Y. (2010). Data-driven predictive control for blast furnace ironmaking process. *Computers & Chemical Engineering*, *34*(11), 1854–1862.