System theory without transfer functions and state-space? Yes, it's possible!

Ivan Markovsky

***Abstract*— The paper demonstrates the claim in the title using missing data estimation as a generic example. The missing data estimation problem includes simulation, Kalman smoothing, and linear quadratic control as special cases. The solution method proposed uses an idea from subspace identification: under a persistency of excitation condition, the image of a Hankel matrix constructed from the data is equal to the behavior of the data-generating system. This fact allows construction of trajectories of the system directly from observed raw data. The construction of trajectories is the key for solving analysis, signal processing, and control problems without parametric model identification. The resulting methods require solution of systems of linear equations, however, the data is assumed exact and obtained from a linear time-invariant system.**

I. Introduction

System theory and the related fields of signal processing and control are currently going through a paradigm shift. The new *data-driven paradigm* aims to achieve a direct map from data to desired result—filtered, predicted, or control signal—without identification of a parametric model of the data generating process. This paper explains how the behavioral paradigm [1], [2] contributed to the data-driven paradigm and how they are both related to low-rank approximation [3].

The key feature that distinguishes the behavioral from the classical and modern paradigms is that a dynamical system is defined as a set of signals, called the behavior. This shift of perspective led to a result that gives sufficient conditions under which the image of a Hankel matrix constructed from data coincides with the behavior of a discrete-time linear time-invariant system [4]. Due to its importance the result became known as the *fundamental lemma.* Under the conditions of the fundamental lemma, the image of a Hankel matrix is a nonparametric representation of the data-generating system. In the last years generalizations and alternative proofs of the fundamental lemma appeared [5], [6], [7], [8].

The fundamental lemma and the Hankel matrix representation opened the path to system theoretic interpretation of the N4SID and MOESP subspace identification methods [9], [10], new identification methods [11], data-driven simulation and linear-quadratic tracking control methods [12]. The method of [12] for linear-quadratic tracking yields an open-loop solution. Combined with receding horizon predictive control, it led to a practical control procedure called *data-enabled predictive control (DeePC)* [13], [14], [15], [16].

Data-driven methods based on the Hankel matrix representation have the following desirable features:

- *generality*—they are applicable for any high-order multivariable linear time-invariant system;
- *simplicity*—require concepts and methods from the behavioral system theory and linear algebra only;
- *practicality*—lead directly to algorithms that are easily implementable in practice.

In comparison, model-based methods require choice of model representation, identification and validation. Depending on the choice of the model representation, the method may have limited generality, *e.g.*, single-input single-output, finite impulse response, and zero initial conditions. If the method is generalizable to multivariable infinite impulse response systems and nonzero initial conditions, the generalizations may require nontrivial modifications and increase the complexity. The complexity of a model-based method is further increased by the complexity of the identification method, used to obtain the model. Finally, the model representation obtained by an identification method may be different from the one used by the model-based method. As a result, the model uncertainty delivered by the identification method may be incompatible with the one used by the model-based design method. For example, the identification method may deliver a confidence ellipsoid for a transfer function model parameters while the design method may be based on $\mathscr{H}_\infty$ robust design for a state-space model. These and other issues related to model-based methods motivate the shift to the data-driven paradigm, which provides an end-to-end solution.

In order to illustrate the claim that the data-driven methods are general, simple, and practical, we consider the problem of *missing data estimation*—given some elements of a trajectory of a dynamical system, find the missing ones. As shown in [17], [18] the missing data estimation problem is generic and includes as special cases various signal processing and control problems, such as simulation, Kalman smoothing, and linear quadratic tracking control. The data-driven method presented uses a complete trajectory of the data-generating system without identifying a parametric model of the system.

In solving the missing data estimation problem, we assume that the given complete trajectory is exact. The resulting missing data recovery problem, called *data-driven dynamic interpolation*, is solved in [17]. The issue of dealing with noisy data is not considered in this paper. It leads to mosaic-Hankel structured low-rank approximation and completion problem [3]. Since this is a nonconvex optimization problem, convex relaxation techniques are used for its solution.

Outline of the paper: Section II shows a Matlab demo. The reader can copy and paste the code from the paper into the Matlab's command line, reproducing the results. Section III, explains how and under what conditions the method used in the demo recovers exactly the missing data. Conclusions and directions for future work are given in Section IV.

Vrije Universiteit Brussel, Department ELEC, Pleinlaan 2, 1050 Brussels, Belgium (email: `imarkovs@vub.be`).

II. Matlab demo

This section presents a Matlab demo of a generic data-driven method. The problem setup given in Section II-A is recovery of missing values of a linear time-invariant system's trajectory. The method is illustrated in Section II-B for recovery of a partially specified trajectory where the given elements are randomly sampled in time. Section II-C shows how the method solves also the simulation problem where the given elements are the initial conditions and the rest of the trajectory is the to-be-recovered missing data. Finally, Section II-D shows how the same method can be used also for noise filtering, where the whole trajectory is given but it is corrupted by zero mean additive noise.

A. Setup

We start by defining a linear time-invariant system sys (the data-generating system) and simulating a trajectory wd of sys (the data trajectory):

```
n = 6; Td = 100;
z = [1.0110 * exp(i * 1.0518)
     1.0128 * exp(i * 0.5207)
     1.0085 * exp(i * 0.0066)];
sys = ss(zpk([], [z; conj(z)], 1, 1));
wd  = initial(sys, rand(n, 1), Td-1);
```

The system is of order n=6 and is defined by its poles. It is unstable and is represented in Matlab by the state-space object sys. The trajectory wd has Td=100 samples. It is generated from initial conditions only (free response).

Another T=30 samples long free response trajectory w of sys is simulated and ng=10 samples are randomly selected:

```
T = 30; ng = 10;
x0 = [100 10 -7 -1 .5 -1.5]';
w = initial(sys, x0, T-1);
t_given = randperm(T, ng);
w_given = w(t_given);
```

The problem considered is to reconstruct w, using the data trajectory wd and the ng given samples w_given of w. Note that the system sys and its order n are not given. The only prior knowledge about the data-generating system sys used by the method is that it is linear time-invariant.

B. Random samples

The solution method presented next does not identify the data-generating system sys from the data trajectory wd. Instead it computes an estimate wh of w directly from wd and w_given. The method leads to the following algorithm:

1) construct a Hankel matrix H from wd,
2) solve the system of linear equations

   g = H(t_given, :) \ w_given,

3) define wh = H * g.

It is implemented in the following two lines of Matlab code:

```
H  = hankel(wd(1:T), wd(T:end));
wh = H * (H(t_given, :) \ w_given);
```

The estimate of the missing values w(t_missing), where

```
t_missing = setdiff(1:T, t_given);
```

is wh(t_missing). In the simulation example, the reconstruction of w(t_missing) is exact up to numerical errors due to the finite precision arithmetic:

```
norm(w - wh) % -> 2.1405e-12
```

The signals w, wh, and the given samples w_given are shown in Figure 1.

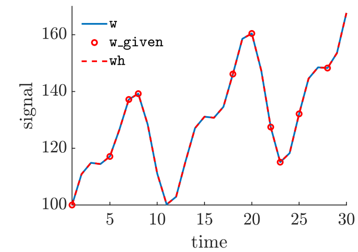

Fig. 1. Reconstruction of a trajectory w of a 6th order single-output linear time-invariant system from randomly selected samples w_given. The trajectory wh reconstructed by the data-driven method matches w.

C. Simulation

In Section II-A, the trajectory w is simulated by the function initial of Matlab, using the state-space representation sys. Next, we show an alternative data-driven simulation method: w is obtained from the first n samples of w (initial condition) and the data trajectory wd. The solution is given by the missing data recovery method of Section II-B:

```
t_given = 1:n; w_given = w(t_given);
wh = H * (H(t_given, :) \ w_given);
norm(w - wh) % -> 2.8294e-11
```

Indeed, simulation is a special case of missing data recovery, so that a new method is not needed. The result is shown in Figure 2, top. Alternatively, w can be recovered from its last n samples (terminal conditions):

```
t_given = T-n+1:T; w_given = w(t_given);
wh = H * (H(t_given, :) \ w_given);
norm(w - wh) % -> 1.8593e-10
```

The result is shown in Figure 2, bottom.

D. Smoothing

In this section, we consider a signal from noise separation problem. The true (noise free) signal w0 is a trajectory of a linear time-invariant system sys and the noise wn is a stochastic process. The errors-in-variables Kalman smoother [19] minimizes the ℓ_2-norm of the approximation error norm(w - wh), where w = w0 + wn is the noisy data and wh is a trajectory of sys. The Kalman smoother assumes that a state space representation of the system sys is a priori given. In this section, we consider the data-driven version of this problem: as in Sections II-B and II-C, instead of sys we are given the trajectory wd of sys.

In the simulation example, we add zero-mean white Gaussian noise to the trajectory w

```
w0 = w; wn = randn(T, 1);
w = w0 + 0.1 * norm(w0) * wn / norm(wn);
```

so that, now, w is a "noisy signal" and w0 is the true value of w. The goal of the smoothing problem is to estimate w0 from

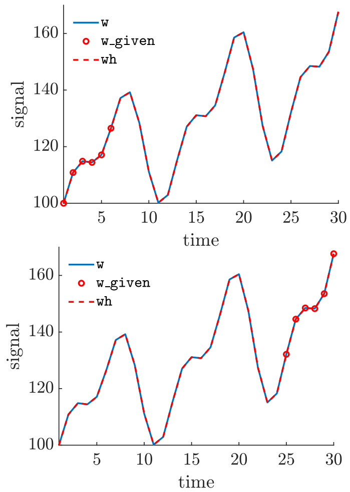

Fig. 2. Simulation of a trajectory `w` from initial conditions `w(1:n)` (top) and terminal conditions `w(end-n+1:end)` (bottom). The simulated trajectories `wh` by the data-driven method match `w` exactly.

`w` and the prior knowledge that `w0` is a trajectory of `sys`. The data-driven smoothing problem is a direct map from `w` and `wd` to an estimate `wh` of `w0`.

The data-driven smoothing problem can be solved again by the method of Section II-B. The only modification needed is to replace the exact solution of the system of linear equations (the backslash operator \ in Matlab) by an approximate solution in the least-squares sense (the `pinv` function):

```
wh = H * pinv(H) * w;
```

The results are shown in Figure 3. We verify that there is indeed a reduction of the noise-to-signal ratio:

```
[norm(w0 - w) norm(w0 - wh)] / norm(w0)
% -> 0.1000     0.0399
```

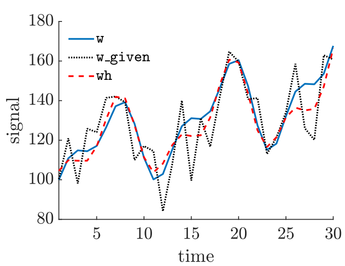

Fig. 3. Smoothing of a noisy trajectory `w`. The smoothed trajectory `wh` obtained by the data-driven method is an approximation of the true value `w0`. Visually `wh` is a better approximation of `w0` than the noisy data `w`.

E. Summary

Using the data-driven method it is possible to solve general signal processing problems where some of the data is exact, some is noisy, and some is missing. The exact data is interpolated, the noisy data is smoothed, and the missing data is estimated. The data-driven smoothing method presented in Section II-D can deal with missing and noisy data. The method is implemented by the following Matlab code:

```
H  = hankel(wd(1:T), wd(T:end));
wh = H * pinv(H) * w;
```

III. Derivation of the method

A. Notation and preliminary results

The key advantage of adopting the behavioral setting in data-driven signal processing and control is that a dynamical system is defined abstractly as a set of trajectories without a priori reference to a parametric representation of this set. In this paper, we consider linear time-invariant systems, so that the models are shift-invariant subspaces. The notation used in the rest of the paper is summarized in Table I.

TABLE I
Summary of the notation.

$w \in (\mathbb{R}^q)^{\mathbb{N}}$	q-variate discrete-time signals $w : \mathbb{N} \to \mathbb{R}^q$
$w\|_L$	restriction of w to the interval $[1, L]$, *i.e.*, $w\|_L := \big(w(1), \dots, w(L)\big)$
σ	the shift operator, $(\sigma w)(t) := w(t+1)$
$\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{N}}$	discrete-time dynamical system $\mathscr{B}$
$\mathscr{L}^q$	linear time-invariant systems with q variables
$\boldsymbol{m}(\mathscr{B})$	number of inputs of $\mathscr{B}$
$\boldsymbol{\ell}(\mathscr{B})$	lag of $\mathscr{B}$
$\boldsymbol{n}(\mathscr{B})$	order of $\mathscr{B}$
$\boldsymbol{c}(\mathscr{B})$	complexity of $\mathscr{B}$, $\boldsymbol{c}(\mathscr{B}) := \big(\boldsymbol{m}(\mathscr{B}), \boldsymbol{\ell}(\mathscr{B}), \boldsymbol{n}(\mathscr{B})\big)$
$\mathscr{L}^q_c$	$:= \{\, \mathscr{B} \in \mathscr{L}^q \mid \boldsymbol{c}(\mathscr{B}) \le c \,\}$
$\mathscr{H}_L(w)$	Hankel matrix with L block rows constructed from w

Associated with a linear time-invariant system $\mathscr{B}$ are the natural numbers $\boldsymbol{m}(\mathscr{B})$—number of inputs, $\boldsymbol{\ell}(\mathscr{B})$—lag, and $\boldsymbol{n}(\mathscr{B})$—order. They are defined as properties of the system and not as properties of its representations [1]. 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 the linearity of $\mathscr{B}$, $\mathscr{B}|_L$ is a subspace of $\mathbb{R}^{qL}$. Its dimension $\dim \mathscr{B}|_L$ is [7]

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

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

$$\mathscr{H}_L(w_{\rm d}) := \begin{bmatrix} w_{\rm d}(1) & w_{\rm d}(2) & \cdots & w_{\rm d}(T_{\rm d}-L+1) \\ w_{\rm d}(3) & w_{\rm d}(4) & \cdots & w_{\rm d}(T_{\rm d}-L+3) \\ \vdots & \vdots & & \vdots \\ w_{\rm d}(L) & w_{\rm d}(L+1) & \cdots & w_{\rm d}(T_{\rm d}) \end{bmatrix}.$$

The columns of $\mathscr{H}_L(w_{\rm d})$ are vectors in $\mathbb{R}^{qL}$. Viewed as L-samples long signals, they are trajectories of $\mathscr{B}$. Combining this fact with (1), it follows that the rank of $\mathscr{H}_L(w)$ is bounded by $\dim \mathscr{B}|_L$. The result extends to multiple trajectories. Let $w_{\rm d} \in \mathscr{B}|_{T_{\rm d}}$ and $w \in \mathscr{B}|_T$ with $T_{\rm d} \ge T \ge \boldsymbol{\ell}(\mathscr{B})$,

$$\operatorname{rank} \begin{bmatrix} \mathscr{H}_L(w_{\rm d}) & \mathscr{H}_L(w) \end{bmatrix} \le \boldsymbol{m}(\mathscr{B})L + \boldsymbol{n}(\mathscr{B}), \quad \text{for any } L \in [\boldsymbol{\ell}(\mathscr{B}), T]. \tag{2}$$

Equation (2) is a link between system theory (trajectories of a linear time-invariant system) and linear algebra (rank deficiency of a matrix). It allows us to pose problems for linear time-invariant systems without involving a parametric model representation. The solution is obtained directly from data by low-rank matrix completion and approximation.

For any $w_\mathrm{d} \in \mathscr{B}|_{T_\mathrm{d}}$ and $L \in [1, T_\mathrm{d}]$, we have that $\mathscr{B}|_L \subseteq \operatorname{image} \mathscr{H}_L(w_\mathrm{d})$. However, under the condition

$$\operatorname{rank} \mathscr{H}_L(w_\mathrm{d}) = \boldsymbol{m}(\mathscr{B})L + \boldsymbol{n}(\mathscr{B}), \quad L > \boldsymbol{\ell}(\mathscr{B}) \tag{3}$$

equality holds, *i.e.*, $\mathscr{B}|_L = \operatorname{image} \mathscr{H}_L(w_\mathrm{d})$. Then, the constraint $w' \in \mathscr{B}|_L$ is equivalent to existence of a solution $g \in \mathbb{R}^{T_\mathrm{d}-L+1}$ of the system $\mathscr{H}_L(w_\mathrm{d})g = w'$.

B. Problem formulation

Consider the signal $w \in (\mathbb{R}^q)^T$. For a vector of indices $I \in \{1, \ldots, qT\}^K$, we define $w|_I := \begin{bmatrix} w_{I_1} & \cdots & w_{I_K} \end{bmatrix}^\top \in \mathbb{R}^K$ as the subvector of $w \in \mathbb{R}^{qT}$ with indices I. Similarly, $\mathscr{H}_T(w_\mathrm{d})|_I$ is the submatrix of $\mathscr{H}_T(w_\mathrm{d})$ with row indices I. In what follows, I_given denotes the indices of the *given elements* of w. The set of indices I_missing of the missing elements of w is then the set difference of $\{1, \ldots, qT\}$ and I_given.

The missing data estimation problem is defined as follows: Given a trajectory $w_\mathrm{d} \in \mathscr{B}|_{T_\mathrm{d}}$ of a linear time-invariant system $\mathscr{B} \in \mathscr{L}^q$ and a partially specified trajectory $w|_{I_\text{given}}$, $w \in \mathscr{B}|_T$,

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{w} \quad \|w|_{I_\text{given}} - \widehat{w}|_{I_\text{given}}\| \\ &\text{subject to} \quad \widehat{w} \in \mathscr{B}|_T. \end{aligned} \tag{4}$$

Theorem 1. *Assuming that (3) holds, the missing data data-driven estimation problem (4) is equivalent to*

$$\begin{aligned} &\textit{minimize} \quad \textit{over } \widehat{w} \quad \|w|_{I_\textit{given}} - \widehat{w}|_{I_\textit{given}}\| \\ &\textit{subject to} \quad \textit{rank} \begin{bmatrix} \mathscr{H}_L(w_\mathrm{d}) & \mathscr{H}_L(\widehat{w}) \end{bmatrix} \leq \mathtt{m}L + \mathtt{n}. \end{aligned} \tag{5}$$

Problem (5) is mosaic-Hankel structured low-rank matrix completion and approximation [20].

C. Solution method

Due to the rank constraint (5) is a nonconvex optimization problem. A convex relaxation is given by regularization with the nuclear norm ($\|\cdot\|_*$):

$$\begin{aligned} \text{minimize} \quad \text{over } \widehat{w} \quad & \|w|_{I_\text{given}} - \widehat{w}|_{I_\text{given}}\| \\ & + \gamma \left\| \begin{bmatrix} \mathscr{H}_L(w_\mathrm{d}) & \mathscr{H}_L(\widehat{w}) \end{bmatrix} \right\|_*. \end{aligned}$$

Another approach for solving (4), used in [17], is to take $L = T$. Then, the constraint of (5) is $\widehat{w} = \mathscr{H}_T(w_\mathrm{d})g$ and problem (5) becomes a standard least-squares problem:

$$\text{minimize} \quad \text{over } g \quad \|w|_{I_\text{given}} - \mathscr{H}_T(w_\mathrm{d})|_{I_\text{given}} g\|, \tag{6}$$

which solution is given in closed form by the pseudo-inverse:

$$\widehat{w} = \mathscr{H}_L(w_\mathrm{d}) \big(\mathscr{H}_L(w_\mathrm{d})|_{I_\text{given}} \big)^+ w|_{I_\text{given}}. \tag{7}$$

Note 2 (Simultaneous missing data estimation and interpolation). A generalization of the data-driven missing data approximation problem (4) is to add equality constraints in order to achieve exact interpolation of certain specified data points. Up to $\operatorname{rank} \mathscr{H}_T(w_\mathrm{d})|_{I_\text{given}}$ equality constraints can be added as exact interpolation conditions, while retaining feasibility. Note that $\operatorname{rank} \mathscr{H}_T(w_\mathrm{d})|_{I_\text{given}} \leq \boldsymbol{m}(\mathscr{B})T + \boldsymbol{n}(\mathscr{B})$. The resulting problem is equality constrained least-squares minimization, so that it is still convex.

IV. Conclusions

The behavioral paradigm defines a dynamical system as a set of trajectories, thus decoupling it from its parametric representations by equations. For an exact trajectory of a discrete-time linear time-invariant system, under the persistency of excitation condition (3), the image of a Hankel matrix constructed from the data coincides with the restricted behavior of the system. We illustrated the generality, simplicity, and practicality of a data-driven method for missing data estimation and approximation. The method applies to general linear time-invariant systems and has no hyper parameters. It requires a solution of a system of linear equations only. Efficient implementation is a topic of current research.

Acknowledgements

The research leading to these results has received funding from: ERC Grant 258581, FWO projects G028015N and G090117N; and FNRS–FWO EOS Project 30468160.

References

[1] J. C. Willems, “From time series to linear system—Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling,” *Automatica*, vol. 22, 23, pp. 561–580, 675–694, 87–115, 1986, 1987.

[2] J. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory*. Springer-Verlag, 1998.

[3] I. Markovsky, *Low-Rank Approximation: Algorithms, Implementation, Applications*, 2nd ed. Springer, 2019.

[4] J. C. Willems, P. Rapisarda, I. Markovsky, and B. De Moor, “A note on persistency of excitation,” *Control Lett.*, vol. 54, pp. 325–329, 2005.

[5] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi, “Willems’ fundamental lemma for state-space systems and its extension to multiple datasets,” *Control Systems Letters*, vol. 4, pp. 602–607, 2020.

[6] J. Rueda-Escobedo and J. Schiffer, “Data-driven internal model control of second-order discrete Volterra systems,” arXiv:2003.14158, 2020.

[7] I. Markovsky and F. Dörfler, “Identifiability in the behavioral setting,” Vrije Univ. Brussel, Tech. Rep., 2020, http://homepages.vub.ac.be/~imarkovs/publications/identifiability.pdf.

[8] ——, “Behavioral systems theory in data-driven analysis, signal processing, and control,” Vrije Univ. Brussel, Tech. Rep., 2021. http://homepages.vub.ac.be/~imarkovs/publications/overview-ddctr.pdf.

[9] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory, implementation, applications*. Boston: Kluwer, 1996.

[10] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, 2007.

[11] I. Markovsky, J. C. Willems, P. Rapisarda, and B. De Moor, “Algorithms for deterministic balanced subspace identification,” *Automatica*, vol. 41, no. 5, pp. 755–766, 2005.

[12] I. Markovsky and P. Rapisarda, “Data-driven simulation and control,” *Int. J. Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[13] J. Coulson, J. Lygeros, and F. Dörfler, “Data-enabled predictive control: In the shallows of the DeePC,” in *European Control Conf.*, 2019, pp. 307–312.

[14] J. Berberich, *et al.*, “Data-driven model predictive control with stability and robustness guarantees,” *IEEE Trans. Automat. Contr.*, 2020.

[15] A. Xue and N. Matni, “Data-driven system level synthesis,” *arXiv:2011.10674*, 2020.

[16] M. Yin, A. Iannelli, and R. Smith, “Maximum likelihood estimation in data-driven modeling and control,” *arXiv:2011.00925*, 2020.

[17] I. Markovsky and F. Dörfler, “Data-driven dynamic interpolation and approximation,” Vrije Univ. Brussel, Tech. Rep., 2021, http://homepages.vub.ac.be/~imarkovs/publications/ddint.pdf.

[18] Z.-S. Hou and Z. Wang, “From model-based control to data-driven control: Survey, classification and perspective,” *Information Sciences*, vol. 235, pp. 3–35, 2013.

[19] I. Markovsky and B. De Moor, “Linear dynamic filtering with noisy input and output,” *Automatica*, vol. 41, no. 1, pp. 167–171, 2005.

[20] I. Markovsky, “A missing data approach to data-driven filtering and control,” *IEEE Trans. Automat. Contr.*, vol. 62, pp. 1972–1978, 2017.