ETH zürich

ETH Library

Maximum Likelihood Estimation in Data-Driven Modeling and Control

Journal Article

Author(s):
Yin, Mingzhou; Iannelli, Andrea; Smith, Roy

Publication date:
2023-01

Permanent link:
https://doi.org/10.3929/ethz-b-000525037

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
IEEE Transactions on Automatic Control 68(1), https://doi.org/10.1109/TAC.2021.3137788

Funding acknowledgement:
178890 - Modeling, Identification and Control of Periodic Systems in Energy Applications (SNF)

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

Maximum Likelihood Estimation in Data-Driven Modeling and Control

Mingzhou Yin, Andrea Iannelli, and Roy S. Smith, *Fellow, IEEE*

M. Yin, A. Iannelli and R. S. Smith, "Maximum Likelihood Estimation in Data-Driven Modeling and Control," in *IEEE Transactions on Automatic Control*, doi: 10.1109/TAC.2021.3137788.

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

***Abstract*— Recently, various algorithms for data-driven simulation and control have been proposed based on the Willems' fundamental lemma. However, when collected data are noisy, these methods lead to ill-conditioned data-driven model structures. In this work, we present a maximum likelihood framework to obtain an optimal data-driven model, the signal matrix model, in the presence of output noise. Data compression and noise level estimation schemes are also proposed to apply the algorithm efficiently to large datasets and unknown noise level scenarios. Two approaches in system identification and receding horizon control are developed based on the derived optimal estimator. The first one identifies a finite impulse response model. This approach improves the least-squares estimator with less restrictive assumptions. The second one applies the signal matrix model as the predictor in predictive control. The control performance is shown to be better than existing data-driven predictive control algorithms, especially under high noise levels. Both approaches demonstrate that the derived estimator provides a promising framework to apply data-driven algorithms to noisy data.**

***Index Terms*— Data-driven modeling, maximum likelihood estimation, model predictive control, system identification.**

I. Introduction

FOLLOWING its remarkable success in artificial intelligence, learning from data is becoming a popular topic in various engineering domains [1]. This concept is by no means a new idea for control engineering. The system identification paradigm has been widely used in control applications, where data are used to fit an a priori parametrized model [2]. The control strategy is then designed with the identified nominal model based on the certainty equivalence principle.

However, this conventional scheme of learning dynamical systems is challenged by increasing complexity of systems and the large amount of data available. In particular, a low-dimensional model structure that is suitable to design compact, closed-form control strategies can be very hard and costly to obtain for complex systems [3]. Therefore, alternative paths are investigated to facilitate control design directly from raw measurement data of dynamical systems. For example, reinforcement learning techniques are widely applied in this area [4]. Such approaches typically avoid predicting the behavior of systems explicitly but aim at the control strategy directly.

In this work, the conventional parametric model is replaced by a data-driven predictor with a non-parametric structure [5]. In the seminal work from Willems *et al.* [6], a single input-output trajectory of the linear system is shown to be able to characterize all possible trajectories of length up to the order of persistency of excitation by constructing Hankel matrices from data. This result is known as the Willems' fundamental lemma. With this result, the behavior of the system can be simulated and thus controlled by selecting a suitable combination of sections from the known trajectory that satisfies the initial condition constraints [7]–[9].

This observation is especially suitable for optimal trajectory tracking. In this regard, model predictive control (MPC) is known to be very effective when an accurate model of the system is available [10]. From the Willems' fundamental lemma, the output prediction step in MPC can be achieved by using known trajectories of the system directly, instead of an explicit model. This data-driven alternative to MPC algorithms, known as data-enabled predictive control (DeePC) [11], has lead to multiple successful applications [12]–[14] with stability and robustness proofs [15]. This framework is also able to handle online data and parameter variations [16].

These types of "data-driven" approaches differ significantly from model-free approaches, and act as a surrogate for conventional models in model-based control to provide a description of system trajectories based on measured data. In fact, with a low-rank approximation, this approach directly leads to the intersection algorithm in subspace identification where state-space models can be derived [17]. The main differences of the data-driven approach compared to conventional model-based methods are: 1) the model is implicit with no closed-form solution in general; 2) the model is over-parametrized in that it does not impose any assumption on the system structure other than linearity. In this paper, this implicit and over-parametrized model is called the data-driven model.

This work was supported by the Swiss National Science Foundation under Grant 200021_178890.

The authors are with the Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH Zürich), 8092 Zurich, Switzerland (e-mail: myin@control.ee.ethz.ch; iannelli@control.ee.ethz.ch; rsmith@control.ee.ethz.ch).

However, it is well-known that when data are noisy, over-parametrized models may lead to high variances and overfitting [18]. In data-driven modeling, finding a combination of known trajectory sections that give reliable prediction is an ill-conditioned problem for datasets with stochastic noise. In current data-driven control schemes, empirical regularizers [11], [15] or least-norm problems [12], [19], [20] are introduced to select a reasonable combination for prediction. Yet, it is not clear what is the optimal way to combine a large set of known trajectory sections to achieve the most reliable prediction.

Another application of data-driven modeling is to simulate the system response [21], [22]. The main advantage of applying this approach in system identification is that it gives the correct estimation of nonparametric models in the noise-free case. Again in this scenario, the best practice for solving the underdetermined linear system in the Willems' fundamental lemma in the noisy case is not understood. For computational simplicity, the Moore-Penrose pseudoinverse solution that solves the least-norm problem is often the default choice [12], leading to the data-driven subspace predictor [20].

As can be seen from the above discussion, one of the central questions in data-driven approaches based on the Willems' fundamental lemma is how to obtain an optimal data-driven model from a large noise-corrupted dataset [23]. Therefore, in the first part of the paper, we propose a maximum likelihood estimation (MLE) framework to estimate such an optimal model with noise in both offline data and online measurements. This optimal model is named the signal matrix model (SMM). This framework optimizes the combination of offline trajectories by maximizing the conditional probability of observing the predicted output trajectory and the measured past outputs. The SMM is shown to obtain more accurate output estimates than the least-norm solution. In addition, a preconditioning strategy is proposed based on singular value decomposition (SVD) to compress the data matrix such that online complexity is fixed for large datasets. When the noise levels are unknown, they can also be estimated with a data-driven approach.

In the second part of the paper, we present two scenarios where the SMM leads to effective algorithms: 1) estimating finite impulse response (FIR) models in system identification; and 2) obtaining a tuning-free data-driven predictive control scheme. In the first scenario, the impulse response is conventionally estimated by least-squares regression, which requires knowing the input history and neglecting truncation errors. In this work, it is replaced by the signal matrix model simulated with an impulse, which guarantees an unbiased estimate. Results show that the model fitting is enhanced when the transient response is unknown or the truncation error of the impulse response is large.

In the second scenario, we replace the prediction part in the DeePC algorithm with the SMM. This predictor is shown to be superior to the pseudoinverse subspace predictor in predictive control. The main advantage of the proposed algorithm is that it avoids the difficult hyperparameter tuning problem in regularized DeePC. The control performance of the proposed algorithm is shown to be better than the DeePC algorithm with optimal hyperparameters when the noise is significant, and similar in the low noise scenario.

The remainder of the paper is organized as follows. Section II defines the notions and preliminaries used in the paper. Section III reviews the Willems' fundamental lemma and its application to deterministic systems. Section IV derives the signal matrix model with MLE and presents an optimal data-driven simulation algorithm. Section V discusses the use of SMM for large datasets and unknown noise levels and analyzes its performance. This model is then applied to two problems: Section VI identifies an FIR model using SMM simulation; Section VII applies the SMM predictor in predictive control. Section VIII concludes the paper.

II. Notation & Preliminaries

For a vector x, the weighted l_2-norm $(x^\mathsf{T} P x)^{\frac{1}{2}}$ is denoted by $\|x\|_P$. The symbol $\mathcal{N}(\mu, \Sigma)$ indicates a Gaussian distribution with mean μ and covariance Σ. The expectation and the covariance of a random vector x are denoted by $\mathbb{E}(x)$ and $\mathrm{cov}(x)$ respectively. For a matrix X, the vectorization operator stacks its columns in a single vector and is denoted by $\mathrm{vec}(X)$; $X^\dagger$ indicates the Moore-Penrose pseudoinverse; $(X)_{i,j}$ denotes the (i,j)-th entry of X. The symbol $\mathbb{S}^n_{++}$ indicates the set of n-by-n positive definite matrix. For a sequence of matrices $X_1, \dots, X_n$, we denote $[X_1^\mathsf{T} \ \dots \ X_n^\mathsf{T}]^\mathsf{T}$ by $\mathrm{col}\,(X_1, \dots, X_n)$. Given a signal $x : \mathbb{Z} \to \mathbb{R}^n$, its trajectory from k to $k+N-1$ is denoted as $(x_i)_{i=k}^{k+N-1}$, and in the vector form as $\mathbf{x} = \mathrm{col}(x_k, \dots, x_{k+N-1})$.

Consider a discrete-time linear time-invariant (LTI) system with output noise, given by

$$\begin{cases} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t + Du_t + w_t, \end{cases} \tag{1}$$

where $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$, $y_t \in \mathbb{R}^{n_y}$, $w_t \in \mathbb{R}^{n_y}$ are the states, inputs, outputs, and output noise respectively. The system is denoted compactly by (A, B, C, D). The pair (A, B) is controllable if $[B \ AB \ \dots \ A^{n_x-1}B]$ has full row rank.

The notion of persistency of excitation is defined as follows.

Definition 1: A signal trajectory $(x_i)_{i=0}^{N-1} \in \mathbb{R}^n \times \{0, \dots, N-1\}$ is said to be persistently exciting of order L if the block Hankel matrix

$$X = \begin{bmatrix} x_0 & x_1 & \cdots & x_{M-1} \\ x_1 & x_2 & \cdots & x_M \\ \vdots & \vdots & \ddots & \vdots \\ x_{L-1} & x_L & \cdots & x_{N-1} \end{bmatrix} \in \mathbb{R}^{Ln \times M} \tag{2}$$

has full row rank, where $M = N - L + 1$ [6].

Intuitively, this definition means that sections of length L of the trajectory span $\mathbb{R}^{Ln}$. When used as the input to a linear dynamical system, it can thus excite all the controllable behaviors of the system in a window of length L. A necessary condition of Definition 1 is $N \geq L(n+1) - 1$, which gives a lower bound on the trajectory length.

III. Deterministic Data-Driven Modeling

In this section, we first review the Willems' fundamental lemma and a few related results in a state-space formulation, followed by an overview of deterministic data-driven simulation and control.

A. Willems' Fundamental Lemma

Built on the notion of the persistency of excitation, the Willems' fundamental lemma shows that all the behavior of a linear system can be captured by a single persistently exciting trajectory of the system when no noise is present. This lemma was originally proposed in the context of behavioral system theory [6], [24], where systems are characterized by the subspace that contains all possible trajectories. It was later reformulated in the state-space context [8], [9]. In the state-space formulation, the output trajectory is unique to a particular input trajectory when a sufficiently long past input-output trajectory is specified as the initial condition. The length of the past trajectory should not be shorter than the state dimension. This idea has strong ties with the intersection algorithm in subspace identification [17], where a low-order subspace of the data matrices that corresponds to a low state dimension is sought.

We summarize the available results on data-driven modeling based on the Willems' fundamental lemma for finite-dimensional LTI systems, which are the foundation for the data-driven methods discussed in this paper. These results hold exactly only when the system is noise-free, i.e., $\forall i, w_i = 0$.

Theorem 1: Consider a finite-dimensional LTI system (A, B, C, D). Let $(u_i^d, x_i^d, y_i^d)_{i=0}^{N-1}$ be a triple of input-state-output trajectory of the system. If the pair (A, B) is controllable and the input is persistently exciting of order $(L + n_x)$, then

(a) the matrix

$$\begin{bmatrix} X \\ U \end{bmatrix} := \left[\begin{array}{cccc} x_0^d & x_1^d & \cdots & x_{M-1}^d \\ \hline u_0^d & u_1^d & \cdots & u_{M-1}^d \\ \vdots & \vdots & \ddots & \vdots \\ u_{L-1}^d & u_L^d & \cdots & u_{N-1}^d \end{array}\right] \tag{3}$$

has full row rank (Corollary 2 in [6], Theorem 1(i) in [9], Lemma 1 in [8]);

(b) the pair $(u_i, y_i)_{i=0}^{L-1}$ is an input-output trajectory of the system iff there exists g, such that

$$\left[\begin{array}{c} u_0 \\ \vdots \\ u_{L-1} \\ \hline y_0 \\ \vdots \\ y_{L-1} \end{array}\right] = \begin{bmatrix} U \\ Y \end{bmatrix} g := \left[\begin{array}{cccc} u_0^d & u_1^d & \cdots & u_{M-1}^d \\ \vdots & \vdots & \ddots & \vdots \\ u_{L-1}^d & u_L^d & \cdots & u_{N-1}^d \\ \hline y_0^d & y_1^d & \cdots & y_{M-1}^d \\ \vdots & \vdots & \ddots & \vdots \\ y_{L-1}^d & y_L^d & \cdots & y_{N-1}^d \end{array}\right] g \tag{4}$$

(Theorem 1 in [6], Theorem 1(ii) in [9], Lemma 2 in [8]);

(c) $\operatorname{rank}(\operatorname{col}(U, Y)) = n_x + n_u L$ (Theorem 2 in [17]);

(d) the vector $(y_i)_{i=0}^{L'-1}$ is the unique output trajectory of the system with past trajectory $(u_i, y_i)_{i=-L_0}^{-1}$ and input trajectory $(u_i)_{i=0}^{L'-1}$, where $n_y L_0 \geq n_x$ and $L' = L - L_0$, iff there exists g, such that

$$\left[\begin{array}{c} u_{-L_0} \\ \vdots \\ u_{L'-1} \\ \hline y_{-L_0} \\ \vdots \\ y_{L'-1} \end{array}\right] = \begin{bmatrix} U \\ Y \end{bmatrix} g := \left[\begin{array}{cccc} u_0^d & u_1^d & \cdots & u_{M-1}^d \\ \vdots & \vdots & \ddots & \vdots \\ u_{L-1}^d & u_L^d & \cdots & u_{N-1}^d \\ \hline y_0^d & y_1^d & \cdots & y_{M-1}^d \\ \vdots & \vdots & \ddots & \vdots \\ y_{L-1}^d & y_L^d & \cdots & y_{N-1}^d \end{array}\right] g \tag{5}$$

(Proposition 1 in [7]).

Remark 1: The controllability and the persistency of excitation conditions can be relaxed for the rank condition in part (c) (Corollary 19 in [25]) or requirements on the initial state (Theorem 1 in [26]).

In Theorem 1, parts (a) and (b) state the original Willems' fundamental lemma; part (c) draws the connection between data-driven modeling and subspace identification methods; and part (d) further gives the uniqueness of the trajectory by fixing a sufficiently long past trajectory. Part (d) also allows the formulation to be applied in simulation and predictive control.

B. Deterministic Data-Driven Simulation and Control

In the noise-free case, the system can be simulated solely based on a known trajectory by applying Theorem 1(d) [21]. Define

$$U_p = \begin{bmatrix} u_0^d & u_1^d & \cdots & u_{M-1}^d \\ \vdots & \vdots & \ddots & \vdots \\ u_{L_0-1}^d & u_{L_0}^d & \cdots & u_{M+L_0-2}^d \end{bmatrix} \in \mathbb{R}^{L_0 n_u \times M}, \tag{6}$$

$$U_f = \begin{bmatrix} u_{L_0}^d & u_{L_0+1}^d & \cdots & u_{M+L_0-1}^d \\ \vdots & \vdots & \ddots & \vdots \\ u_{L-1}^d & u_L^d & \cdots & u_{N-1}^d \end{bmatrix} \in \mathbb{R}^{L' n_u \times M}, \tag{7}$$

$$\mathbf{u}_{\text{ini}} = \operatorname{col}(u_{-L_0}, \cdots, u_{-1}) \in \mathbb{R}^{L_0 n_u}, \tag{8}$$

$$\mathbf{u} = \operatorname{col}(u_0, \cdots, u_{L'-1}) \in \mathbb{R}^{L' n_u}, \tag{9}$$

and similarly for Y_p, Y_f, $\mathbf{y}_{\text{ini}}$, and $\mathbf{y}$. Then we interpret (5) as an implicit model of the system trajectory parametrized by g, namely

$$\begin{cases} \mathbf{u} = U_f g, & \text{(10a)} \\ \mathbf{y} = Y_f g, & \text{(10b)} \end{cases}$$

subject to the initial condition requirement

$$\begin{cases} \mathbf{u}_{\text{ini}} = U_p g, & \text{(11a)} \\ \mathbf{y}_{\text{ini}} = Y_p g. & \text{(11b)} \end{cases}$$

Thus, the system can be simulated by means of a two-step approach with g as the intermediate parameter as shown in Algorithm 1. The system identification process is performed online for a particular input by estimating g. This algorithm effectively gives an implicit model of the system in the form of

$$\mathbf{y} = f(\mathbf{u}; \mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}, U_p, U_f, Y_p, Y_f), \tag{12}$$

where U_p, U_f, Y_p, Y_f are offline data that describe the behaviors of the system, and $\mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}$ are online data that estimate the initial condition.

Algorithm 1 Noise-free data-driven simulation [21]

1: **Given:** U_p, U_f, Y_p, Y_f.
2: **Input:** $\mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}, \mathbf{u}$.
3: Solve the linear system

$$\text{col}\left(\mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}, \mathbf{u}\right) = \text{col}\left(U_p, Y_p, U_f\right) g \tag{13}$$

for g.
4: **Output:** $\mathbf{y} = Y_f g$.

Multiple control algorithms have been developed based on this model structure [7], [8]. In this work, we focus on the optimal trajectory tracking problem, which optimizes the following control cost over a horizon of length L' at each time instant t [10]:

$$J_{\text{ctr}}(\mathbf{u}, \mathbf{y}) = \sum_{k=0}^{L'-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right), \tag{14}$$

where $\mathbf{r}$ is the reference trajectory, and Q and R are the output and the input cost matrices respectively [10]. At each time instant, the first entry in the newly optimized input trajectory is applied to the system in a receding horizon fashion.

Algorithm 1 can be applied as the predictor in place of the model-based predictor in conventional MPC algorithms. This leads to the following optimization problem

$$\begin{aligned} \underset{\mathbf{u},\mathbf{y},g}{\text{minimize}} \quad & J_{\text{ctr}}(\mathbf{u}, \mathbf{y}) \\ \text{subject to} \quad & (10), (11), \mathbf{u} \in \mathcal{U}, \mathbf{y} \in \mathcal{Y}, \end{aligned} \tag{15}$$

where $\mathcal{U}$ and $\mathcal{Y}$ are the constraint sets of the inputs and the outputs respectively. Vectors $\mathbf{u}_{\text{ini}}$ and $\mathbf{y}_{\text{ini}}$ are the immediate past input-output measurements online. This method is known as the unregularized DeePC algorithm [11].

IV. Maximum Likelihood Data-Driven Model: Signal Matrix Model

The linear system (13) is highly underdetermined when a large dataset is available. When the data are noise-free, this parameter estimation problem is trivial, where any solution to (13) gives an exact output model of the system, according to Theorem 1(d).

However, the problem of finding the model (12) becomes ill-conditioned when the data are noisy. In this case, Theorem 1(c) is no longer satisfied. Instead, $\text{col}(U, Y)$ has full row rank almost surely. If we still follow Algorithm 1, any output trajectory $\mathbf{y}$ can be obtained by choosing different solutions to (13). In fact, Theorem 1(d) does not hold exactly for the noisy case, so satisfying condition (13) is not guaranteed to be statistically optimal. An empirical remedy to this problem is to use the Moore-Penrose pseudoinverse solution of g, namely

$$g_{\text{pinv}} = \text{col}\left(U_p, Y_p, U_f\right)^\dagger \text{col}\left(\mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}, \mathbf{u}\right), \tag{16}$$

which solves the least-norm problem

$$\underset{g}{\text{minimize}} \ \|g\|_2^2 \quad \text{subject to } (13). \tag{17}$$

This solution is known as the subspace predictor related to the prediction error method [12], [20]. However, this predictor fails to appropriately encode the effects of noise in the data matrices. To the best of our knowledge, there is no existing statistical framework for estimating g. In what follows, we will derive a maximum likelihood estimator of g. As opposed to existing algorithms, this estimator obtains a statistically optimal data-driven model for systems with noise. Since this model is expressed purely in terms of matrices of signal trajectories, we name this model the signal matrix model. For simplicity of exposition, the results in the section are stated for the single-input single-output case, but they seamlessly hold for the multiple-input multiple-output case.

A. Derivation of the Maximum Likelihood Estimator

In this work, we consider a scenario where the output errors are i.i.d. Gaussian noise for both offline and online data, i.e.,

$$y_i^d = y_i^{d,0} + w_i^d, \ (w_i^d)_{i=0}^{N-1} \sim \mathcal{N}(0, \sigma^2 \mathbb{I}), \tag{18}$$

$$\mathbf{y}_{\text{ini}} = \mathbf{y}_{\text{ini}}^0 + \mathbf{w}_p, \ \mathbf{w}_p \sim \mathcal{N}(0, \sigma_p^2 \mathbb{I}). \tag{19}$$

Under this noise model, the equations (10a) and (11a) still hold exactly, but the past output equation (11b) includes noise on both sides, which leads to a total least squares problem. In this work, the maximum likelihood interpretation of the total least squares problem is used [27].

Define

$$\hat{\mathbf{y}} = \begin{bmatrix} \epsilon_y \\ \mathbf{y} \end{bmatrix} = Yg - \begin{bmatrix} \mathbf{y}_{\text{ini}} \\ \mathbf{0} \end{bmatrix}, \tag{20}$$

where $\epsilon_y := Y_p g - \mathbf{y}_{\text{ini}}$ is the residual of the past output relation (11b). Then we want to construct an estimator that maximizes the conditional probability of observing the realization $\hat{\mathbf{y}}$ corresponding to the available data given g. Applying vectorization on Yg in (20), we have

$$\hat{\mathbf{y}} = \left(g^\mathsf{T} \otimes \mathbb{I}\right) \text{vec}(Y) - \begin{bmatrix} \mathbf{y}_{\text{ini}} \\ \mathbf{0} \end{bmatrix}, \tag{21}$$

where we make use of the property of the Kronecker product

$$\text{vec}(ABC) = (C^\mathsf{T} \otimes A)\text{vec}(B). \tag{22}$$

Denote the noise-free version of Y_p and Y_f by Y_p^0 and Y_f^0 respectively. Then for a given g, we have

$$\begin{aligned} \mathbb{E}(\hat{\mathbf{y}}|g) &= \mathbb{E}(Y)g - \begin{bmatrix} \mathbb{E}(\mathbf{y}_{\text{ini}}) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} Y_p^0 g - \mathbf{y}_{\text{ini}}^0 \\ Y_f^0 g \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ Y_f^0 g \end{bmatrix}, \\ \text{cov}(\hat{\mathbf{y}}|g) &= \left(g^\mathsf{T} \otimes \mathbb{I}\right) \Sigma_{yd} \left(g \otimes \mathbb{I}\right) + \begin{bmatrix} \sigma_p^2 \mathbb{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} =: \Sigma_y, \end{aligned} \tag{23}$$

where $\Sigma_{yd} = \text{cov}\left(\text{vec}(Y)\right)$. According to the noise model of $\left(y_i^d\right)$ and accounting for the Hankel structure of Y, we have

$$\left(\Sigma_{yd}\right)_{i,j} = \begin{cases} \sigma^2, & \left(\text{vec}(Y)\right)_i = \left(\text{vec}(Y)\right)_j \\ 0, & \text{otherwise} \end{cases}. \tag{24}$$

Then, Σ_y can be calculated as

$$\left(\Sigma_y\right)_{i,j} = \sigma^2 \sum_{k=1}^{M-|i-j|} g_k g_{k+|i-j|} + \begin{cases} \sigma_p^2, & i = j \le L_0 \\ 0, & \text{otherwise} \end{cases}. \tag{25}$$

where g_k denotes the k-th entry of g. The derivation is given in Appendix I. Thus, due to the linearity of the normal distribution, we have

$$\hat{\mathbf{y}}|g \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{0}\\ Y_f^0 g\end{bmatrix}, \Sigma_y\right), \tag{26}$$

which has the probability density

$$p(\hat{\mathbf{y}}|g) = (2\pi)^{-\frac{L}{2}} \det(\Sigma_y)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\begin{bmatrix}Y_p g - \mathbf{y}_{\text{ini}}\\ Y_f g - Y_f^0 g\end{bmatrix}^\top \Sigma_y^{-1} \begin{bmatrix}Y_p g - \mathbf{y}_{\text{ini}}\\ Y_f g - Y_f^0 g\end{bmatrix}\right). \tag{27}$$

Note that here the true output data matrix Y_f^0 is also unknown, and can be estimated with the maximum likelihood approach. In this way, we are ready to derive the signal matrix model by solving the following optimization problem.

$$\underset{g\in\mathcal{G}, Y_f^0}{\text{minimize}} \; -\log p(\hat{\mathbf{y}}|g, Y_f^0), \tag{28}$$

where $\mathcal{G} = \left\{g \in \mathbb{R}^M \,|\, \text{col}\,(U_p, U_f)\, g = \text{col}\,(\mathbf{u}_{\text{ini}}, \mathbf{u})\right\}$ is the parameter space defined by the known noise-free input trajectory.

Substituting (27) into (28), we have the equivalent optimization problem,

$$\underset{g\in\mathcal{G}, Y_f^0}{\text{minimize}} \; \text{logdet}(\Sigma_y(g)) + \begin{bmatrix}Y_p g - \mathbf{y}_{\text{ini}}\\ Y_f g - Y_f^0 g\end{bmatrix}^\top \Sigma_y^{-1}(g) \begin{bmatrix}Y_p g - \mathbf{y}_{\text{ini}}\\ Y_f g - Y_f^0 g\end{bmatrix}. \tag{29}$$

It is easy to see that the optimal value of Y_f^0 is Y_f regardless of the choice of g. So (29) is equivalent to

$$\underset{g\in\mathcal{G}}{\text{minimize}} \; \text{logdet}(\Sigma_y(g)) + \begin{bmatrix}Y_p g - \mathbf{y}_{\text{ini}}\\ \mathbf{0}\end{bmatrix}^\top \Sigma_y^{-1}(g) \begin{bmatrix}Y_p g - \mathbf{y}_{\text{ini}}\\ \mathbf{0}\end{bmatrix}. \tag{30}$$

In this objective function, the first term indicates how accurate the output estimates are. The second term represents how much the estimate deviates from the past output observations.

B. Iterative Computation of the Estimator

To find a computationally efficient algorithm to solve (30), we relax the problem and solve it with sequential quadratic programming (SQP) [28]. First, the covariance matrix Σ_y is approximated with its diagonal part, denoted by $\bar{\Sigma}_y$, i.e.,

$$\left(\bar{\Sigma}_y\right)_{i,j} = \begin{cases}(\Sigma_y)_{i,j}, & i = j\\ 0, & i \neq j\end{cases}. \tag{31}$$

Remark 2: This approximation holds exactly when the data matrices are constructed by truncating $(u_i^d, y_i^d)_{i=0}^{N-1}$ into sections of length L with no overlap, or using multiple independent trajectories of length L, instead of forming Hankel structures. This construction is known as the Page matrix [29] and it was shown in [9] that similar results to Theorem 1 still hold for Page matrices. The Hankel construction is able to use the data more efficiently, but leads to complex noise correlation, which is reflected in the non-diagonal structure of Σ_y. The comparison between the Hankel construction and the Page construction is, however, beyond the scope of this paper. See [25], [30] for more on this topic.

Remark 3: This approximation gives an upper bound on the log-det terms. According to Hadamard's inequality, since $\Sigma_y \in \mathbb{S}_{++}^L$, we have $\text{logdet}(\Sigma_y(g)) \leq \text{logdet}(\bar{\Sigma}_y(g))$.

In this way, problem (30) is approximated as

$$\underset{g\in\mathcal{G}}{\text{minimize}} \; L' \log\left(\|g\|_2^2\right) + L_0 \log\left(\sigma^2 \|g\|_2^2 + \sigma_p^2\right) + \frac{1}{\sigma^2 \|g\|_2^2 + \sigma_p^2} \|Y_p g - \mathbf{y}_{\text{ini}}\|_2^2. \tag{32}$$

This problem can be readily solved by SQP. For each iteration, the following quadratic programming problem is solved.

$$g^{k+1} = \underset{g}{\text{argmin}} \; \lambda(g^k) \|g\|_2^2 + \|Y_p g - \mathbf{y}_{\text{ini}}\|_2^2 \quad \text{subject to} \quad \begin{bmatrix}U_p\\ U_f\end{bmatrix} g = \begin{bmatrix}\mathbf{u}_{\text{ini}}\\ \mathbf{u}\end{bmatrix}, \tag{33}$$

where $\lambda(g^k) = L'\sigma_p^2 / \|g^k\|_2^2 + L\sigma^2$. The objective function in (32) is approximated by a quadratic function around g^k, making use of the local expansion $\log x \approx \log x_0 + \frac{1}{x_0}(x - x_0)$. The optimality conditions of (33) are:

$$\begin{bmatrix}F(g^k) & U^\top\\ U & \mathbf{0}\end{bmatrix}\begin{bmatrix}g^{k+1}\\ \nu^{k+1}\end{bmatrix} = \begin{bmatrix}Y_p^\top \mathbf{y}_{\text{ini}}\\ \tilde{\mathbf{u}}\end{bmatrix}, \tag{34}$$

where $\tilde{\mathbf{u}} = \text{col}(\mathbf{u}_{\text{ini}}, \mathbf{u})$, $F(g^k) = \lambda(g^k)\mathbb{I} + Y_p^\top Y_p$, and $\nu_{k+1} \in \mathbb{R}^L$ is the Lagrange multiplier. The closed-form solution is thus given by

$$\begin{aligned} g^{k+1} &= \left(F^{-1} - F^{-1}U^\top(UF^{-1}U^\top)^{-1}UF^{-1}\right) Y_p^\top \mathbf{y}_{\text{ini}} + F^{-1}U^\top(UF^{-1}U^\top)^{-1}\tilde{\mathbf{u}}\\ &=: \mathcal{P}(g^k)\mathbf{y}_{\text{ini}} + \mathcal{Q}(g^k)\tilde{\mathbf{u}}. \end{aligned} \tag{35}$$

This algorithm converges to a local minimum of problem (32).

Remark 4: Following similar derivations, this algorithm can be extended to the case where i.i.d. Gaussian input errors also exist in offline and online data,, which leads to an additional input regularization term $\|Ug - \tilde{\mathbf{u}}\|_2^2$ in the iterative algorithm.

C. Maximum Likelihood Data-Driven Simulation

Based on the derived maximum likelihood estimator of g, the step of solving the linear system (13) in Algorithm 1 can be replaced by solving the SQP problem (33). For simulation, the SQP problem can be initialized at the pseudoinverse solution g_{pinv} (16). This leads to the following algorithm for maximum likelihood data-driven simulation.

This algorithm gives the signal matrix model in the form of (12). The approximate maximum likelihood estimator (33) has the same $\|g\|_2^2$-penalization term as the least-norm problem (17). However, the estimate g_{SMM} does not lie in the solution space of the underdetermined system (13). The total least squares structure in (11b) leads to the penalization term $\|Y_p g - \mathbf{y}_{\text{ini}}\|_2^2$ in place of the hard constraint in (13).

Algorithm 2 Maximum likelihood data-driven simulation: the signal matrix model

Given: $U_p, U_f, Y_p, Y_f, \sigma, \sigma_p, \epsilon$.
Input: $\mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}, \mathbf{u}$.
$k \leftarrow 0$, $g^0 \leftarrow g_{\text{pinv}}$ from (16)
repeat
  Calculate g^{k+1} with (35).
  $k \leftarrow k+1$
until $\left\|g^k - g^{k-1}\right\| < \epsilon \left\|g^{k-1}\right\|$
Output: $g_{\text{SMM}} = g^k$, $\mathbf{y} = Y_f g^k$.

V. Practical Aspects and Analysis of the Signal Matrix Model

In this section, we first discuss practical scenarios where the dimension of the signal matrix is very large and the noise level information required in formulating the signal matrix model is unknown. Then, the effectiveness of the proposed maximum likelihood framework is analyzed by numerical comparison and covariance analysis.

A. Preconditioning of Data Matrices

In data-driven applications, it is usually assumed that abundant data are available, i.e., $N \gg L$. Under this scenario, the dimension of the parameter vector $g \in \mathbb{R}^M$, which needs to be optimized online, would be much larger than the length of the predicted output trajectory. This leads to high online computational complexity even to estimate a very short trajectory. On the other hand, only $2L$ independent basis vectors are needed to describe all the possible input-output trajectories of length L. It is possible to precondition the data matrices such that only $2L$ basis trajectories are used.

To do this, we propose the following strategy based on the SVD to compress the data such that the dimension of the parameter vector g is $2L$ regardless of the raw data length. Let $\text{col}(U, Y) = WSV^{\mathsf{T}} \in \mathbb{R}^{2L \times M}$ be the SVD of the data matrix. Define the compressed data matrices $\tilde{U}_p, \tilde{Y}_p \in \mathbb{R}^{L_0 \times 2L}$ and $\tilde{U}_f, \tilde{Y}_f \in \mathbb{R}^{L' \times 2L}$ such that

$$\text{col}\left(\tilde{U}_p, \tilde{U}_f, \tilde{Y}_p, \tilde{Y}_f\right) = WS_{2L} \in \mathbb{R}^{2L \times 2L}, \tag{36}$$

where S_{2L} is the first $2L$ columns of S.

It is shown in the following proposition that Algorithm 2 with compressed data matrices obtains exactly the same output trajectory $\mathbf{y}$ as with raw data matrices.

Proposition 1: Let the simulated trajectories with data matrices (U_p, Y_p, U_f, Y_f) and $(\tilde{U}_p, \tilde{Y}_p, \tilde{U}_f, \tilde{Y}_f)$ from Algorithm 2 be $\mathbf{y}$ and $\tilde{\mathbf{y}}$ respectively. Then we have $\tilde{\mathbf{y}} = \mathbf{y}$.

Proof: Define transformed data matrices $\bar{U}_p$, $\bar{Y}_p$, $\bar{U}_f$, and $\bar{Y}_f$ such that $\text{col}\left(\bar{U}_p, \bar{Y}_p, \bar{U}_f, \bar{Y}_f\right) = WS$. Then the relations between the data matrices are given by

$$\begin{aligned}
\text{col}\left(U_p, Y_p, U_f, Y_f\right) &= \text{col}\left(\tilde{U}_p, \tilde{U}_f, \tilde{Y}_p, \tilde{Y}_f\right) V_{2L}^{\mathsf{T}}, \\
\text{col}\left(U_p, Y_p, U_f, Y_f\right) &= \text{col}\left(\bar{U}_p, \bar{Y}_p, \bar{U}_f, \bar{Y}_f\right) V^{\mathsf{T}}, \\
\text{col}\left(\bar{U}_p, \bar{Y}_p, \bar{U}_f, \bar{Y}_f\right) &= \left[\text{col}\left(\tilde{U}_p, \tilde{U}_f, \tilde{Y}_p, \tilde{Y}_f\right) \ \mathbf{0}\right].
\end{aligned} \tag{37}$$

where V_{2L} denotes the first $2L$ columns of V.

Denote the variables with the compressed data matrices by a tilde, and the variables with the transformed data matrices by a bar. Since $V_{2L}^{\mathsf{T}} V_{2L} = \mathbb{I}$, we have $g_{\text{pinv}} = V_{2L}\, \tilde{g}_{\text{pinv}}$. This leads to $\|g_{\text{pinv}}\|_2^2 = \|\tilde{g}_{\text{pinv}}\|_2^2$, and thus $\lambda(g^0) = \lambda(\tilde{g}^0)$.

Suppose at the k-th iteration, $\lambda(g^k) = \lambda(\tilde{g}^k)$. Due to the orthogonality of V and the sparsity structure of $\bar{U}$ and $\bar{Y}_p$, we have $g^{k+1} = V\tilde{g}^{k+1}$, $\bar{g}^{k+1} = \text{col}\left(\tilde{g}^{k+1}, \mathbf{0}\right)$. This leads to

$$g^{k+1} = V_{2L}\, \tilde{g}^{k+1}, \ \left\|g^{k+1}\right\|_2^2 = \left\|\tilde{g}^{k+1}\right\|_2^2. \tag{38}$$

Thus for all k, we have $g^k = V_{2L}\, \tilde{g}^k$. Therefore, the simulated trajectory satisfies $\mathbf{y} = Y_f g^k = \tilde{Y}_f \tilde{g}^k = \tilde{\mathbf{y}}$. ∎

Remark 5: It can be seen from the proof that $\bar{\Sigma}_y(g) = \bar{\Sigma}_y(\tilde{g})$. So with compressed data matrices, the output trajectory estimate has the same covariance as the raw data matrices when Page matrices are used, and the same diagonal components when Hankel matrices are used.

B. Data-driven Noise Level Estimation

When the noise level σ^2 in the output signal matrix Y is unknown, it can be directly estimated from the singular values of a projected signal matrix. In detail, let $\Pi_U^{\perp} = \mathbb{I} - U^{\top}(UU^{\top})^{-1}U$ span the null space of the noise-free matrix U, and Y^0 be the noise-free version of Y. Then according to the persistency of excitation requirement and Theorem 1(c), $\text{rank}(Y^0\Pi_U^{\perp}) = n_x$, and thus $Y\Pi_U^{\perp}$ is a perturbed rank-n_x matrix. We apply the robust noise level estimator for perturbed low-rank matrices presented in Section III-E of [31]

$$\hat{\sigma}^2 = \frac{s_{\text{med}}^2}{M\mu(L/M)}, \tag{39}$$

where s_{med} is the median of the singular values of $Y\Pi_U^{\perp}$ and $\mu(\beta)$ is the median of the Marchenko-Pastur distribution with aspect ratio β. This estimator compares the perturbed singular values with the ideal distribution of the noise singular values to estimate σ^2. It is applicable when s_{med} comes purely from noise, i.e., $L > 2n_x$.

The noise level of online data σ_p^2 can be set to zero when initial conditions are known exactly, or to σ^2 when the same sensor is used for offline and online measurements. Otherwise, online measurements can be taken beforehand and σ_p^2 can be estimated similarly to σ^2.

C. Comparison of Data-Driven Predictors

The performance of Algorithm 2 is analyzed numerically by comparing the accuracy of the simulated output $\mathbf{y}$ measured by fitting metric

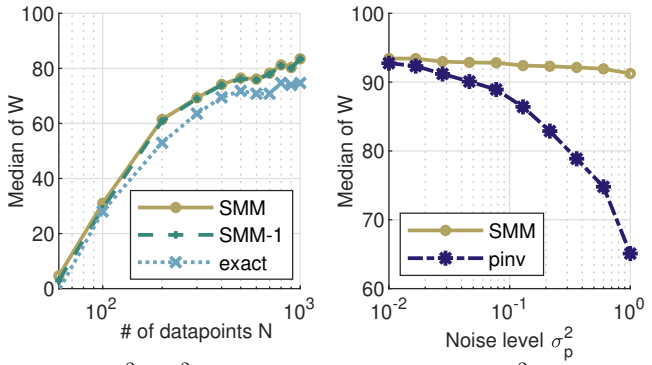
$$W = 100 \cdot \left(1 - \left[\frac{\sum_{i=1}^{L'}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{L'}(y_i - \bar{y})^2}\right]^{1/2}\right), \tag{40}$$

where y_i are the true outputs, $\hat{y}_i$ are the estimated outputs, and $\bar{y}$ is the mean of the true outputs. We compare 1) *pinv*: the least-norm solution (16), 2) *exact*: the SQP solution of problem (30) initialized at g_{pinv}, 3) *SMM-1*: the solution after one iteration of Algorithm 2, and 4) *SMM*: Algorithm 2.

Consider random single-input single-output systems with state dimensions between 2 and 10 (generated by MATLAB

function `drss`). The following parameters are used: $L_0 = n_x$, $L' = 10$. Inputs for the identification data $(u_i^d)_{i=0}^{N-1}$ and simulation conditions $\mathbf{u}_{\text{ini}}$, $\mathbf{y}_{\text{ini}}$, $\mathbf{u}$ are all unit i.i.d. Gaussian. For each analysis, 100 Monte Carlo simulations are conducted.

The simulation accuracy of different MLE algorithms are plotted in Figure 1(a) for different data sizes N. It can be seen that for small data sizes, the *exact* estimate obtains very similar performance to the SMM estimates. This indicates that the approximate solution obtains a close match to the original MLE problem. For larger data sizes, due to the increasing dimension of g, the performance of *exact*, where the data compression scheme does not apply, becomes worse. On the other hand, Algorithm 2 converges very quickly as the one-iteration solution *SMM-1* obtains almost identical performance to the converged solution *SMM* at all data sizes.



(a) $\sigma^2 = \sigma_p^2 = 1$ (b) $N = 100$, $\sigma^2 = 0.01$

Fig. 1. Comparison of simulation accuracy with different data-driven predictors.

The *SMM* estimate is compared against *pinv* in Figure 1(b) for different online noise levels σ_p^2. It is showcased that *SMM* is more accurate than *pinv* due to the inclusion of the correct noise model. In particular, this performance improvement is more significant when σ_p^2 is large. In order to assess the general validity of the results shown in Figure 1(b), it is demonstrated theoretically in the following proposition that the signal matrix model obtains a smaller covariance than the least-norm solution when noise is present only in $\mathbf{y}_{\text{ini}}$.

Proposition 2: Let g_{pinv} and g_{SMM} be the estimates from the least-norm solution (16) and Algorithm 2 respectively. When $\sigma^2 = 0$, we have $\text{tr}(\text{cov}(g_{\text{SMM}})) < \text{tr}(\text{cov}(g_{\text{pinv}}))$.

Proof: See Appendix II. ∎

VI. Impulse Response Estimation With the Signal Matrix Model

We propose here a system identification method that identifies an FIR model of the system by signal matrix model simulation. Numerical tests show that model fitting is improved compared to the conventional least-squares estimate, when the truncation error is large or the input history is unknown.

A. Impulse Response Estimation

Consider the problem of estimating the impulse response model $(h_i)_{i=0}^{\infty}$ of a system from data. The output y_t is given by

$$y_t = \sum_{i=0}^{\infty} h_i u_{t-i}. \tag{41}$$

The conventional approach is to formulate a linear regression to estimate a finite truncation of the impulse response

$$\underbrace{\begin{bmatrix} y_0^d \\ y_1^d \\ \vdots \\ y_{N-1}^d \end{bmatrix}}_{y_N} = \underbrace{\begin{bmatrix} u_0^d & u_{-1}^d & \cdots & u_{1-n}^d \\ u_1^d & u_0^d & \cdots & u_{2-n}^d \\ \vdots & \vdots & \ddots & \vdots \\ u_{N-1}^d & u_{N-2}^d & \cdots & u_{N-n}^d \end{bmatrix}}_{\Phi_N} \underbrace{\begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{n-1} \end{bmatrix}}_{h}, \tag{42}$$

where n is the length of the impulse response to be estimated. The regression problem can then be solved by least squares with the closed-form solution

$$\hat{h}_{\text{LS}} = \left(\Phi_N^{\mathsf{T}} \Phi_N\right)^{-1} \Phi_N^{\mathsf{T}} y_N. \tag{43}$$

There are two main assumptions underlying this formulation: 1) the truncation error of the finite impulse response is negligible, i.e., $h_i \approx 0$ for all $i \geq n$; and 2) additional input measurements $(u_i^d)_{i=1-n}^{-1}$ are available. With these two assumptions, the least-squares solution is known to be the best unbiased estimator with i.i.d. Gaussian output noise [2].

However, these assumptions may not be satisfied in practice. When the internal dynamics matrix A has a large condition number, a very long impulse response sequence is needed to remove the truncation error even for a low-order system. In this case, the least-squares algorithm may become impractical due to limited data length and/or computation power. If the truncation error is not negligible, the estimator is not correct, i.e., in the noise-free case, the estimate does not coincide with the true system. When the input history is unknown, the first $(n-1)$ input measurements have to be used solely for initial condition estimation, in which case the data efficiency is substantially affected when a large n is needed.

In this work, we propose using the signal matrix model to estimate the impulse response by finding the length-n response to a pulse input from zero initial conditions, i.e.,

$$\mathbf{u}_{\text{ini}} = \mathbf{0},\ \mathbf{y}_{\text{ini}} = \mathbf{0},\ \mathbf{u} = \text{col}(1, \mathbf{0}),\ L' = n. \tag{44}$$

Since the initial condition is known exactly, we have $\sigma_p = 0$. Then the output trajectory $\mathbf{y}$ is an estimate of the impulse response h of length n of the system [21]. This approach requires neither of the assumptions for the least-squares method. Instead of requiring a length-$(n-1)$ input history sequence, this approach only uses the first L_0 entries of the data to estimate the initial condition. In fact, the estimator is correct and unbiased for an arbitrary length n and unknown input history as shown in Theorem 1(d), as long as the persistency of excitation condition is satisfied.

B. Numerical Results

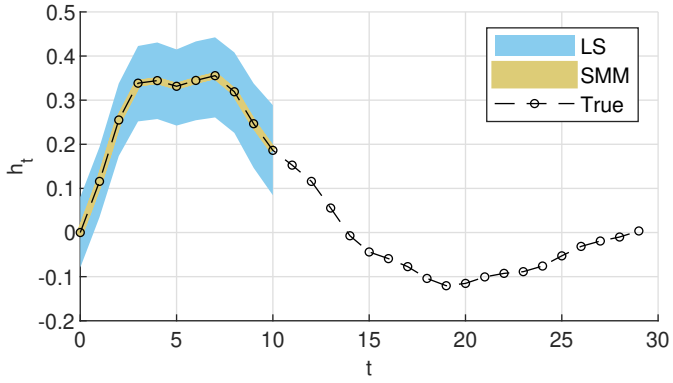
In this subsection, the proposed algorithm is tested against the least-squares estimate by applying it to numerical examples. We compare the proposed signal matrix model estimate *SMM* (Algorithm 2 with (44)) with the least-squares estimate *LS* (43). The parameters used in the simulation are $N =$

50, $L_0 = 4$, $n = L' = 11$, $\sigma^2 = 0.01$. In *SMM*, the noise level σ^2 is estimated using (39). The identification data are generated with unit i.i.d. Gaussian input signals. For each case, 1000 Monte Carlo simulations are conducted.

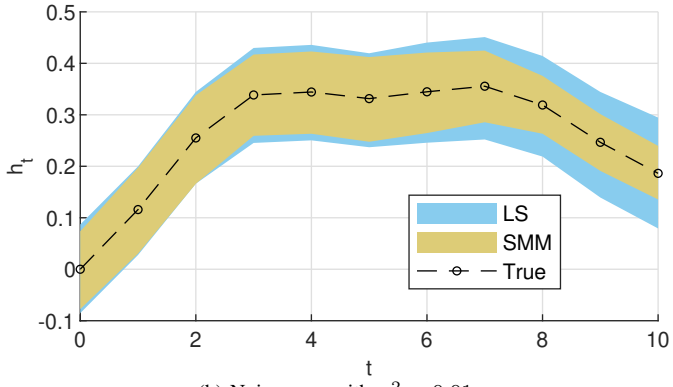
In the first example, we consider the following fourth-order LTI system tested in [32]

$$G_1(z) = \frac{0.1159(z^3 + 0.5z)}{z^4 - 2.2z^3 + 2.42z^2 - 1.87z + 0.7225}. \quad (45)$$

This system is relatively slow. The truncation error is significant when $n = 11$ is selected. First, the *LS* and *SMM* algorithms are compared under the noise-free case, and the results are shown in Figure 2(a). It can be clearly seen that *LS* is not correct due to the presence of truncation errors, whereas the SMM estimator is correct. When the noise is present, the *LS* and *SMM* algorithms are compared in Figure 2(b). The *SMM* estimator have smaller variance compared to *LS*.



(a) Noise-free case



(b) Noisy case with $\sigma^2 = 0.01$

Fig. 2. Comparison of impulse response estimation with truncation errors. Colored area: estimates within two standard deviations.

In the second example, we focus on the effect of unknown input history by investigating a faster LTI system used in [32]

$$G_2(z) = \frac{0.9183z}{z^2 + 0.24z + 0.36}. \quad (46)$$

In this case, the truncation error is already negligible at $n = 11$, but we assume the input history is unknown. The results of the estimation are illustrated in Figure 3. The result of the *SMM* algorithm is shown to be more accurate than the *LS* algorithm, especially for the first four coefficients.

To quantitatively assess the performance of different algorithms, we quantify the model fitting by the metric W (40)

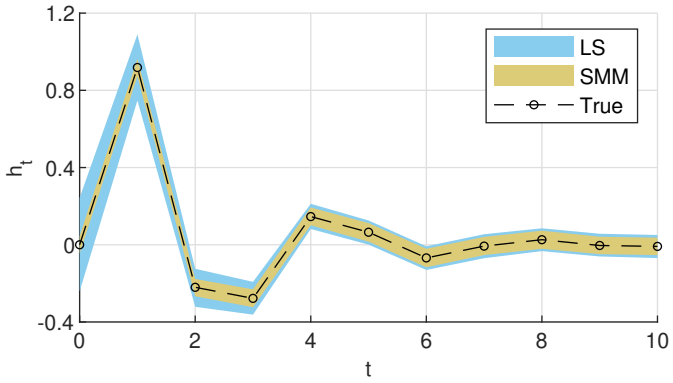
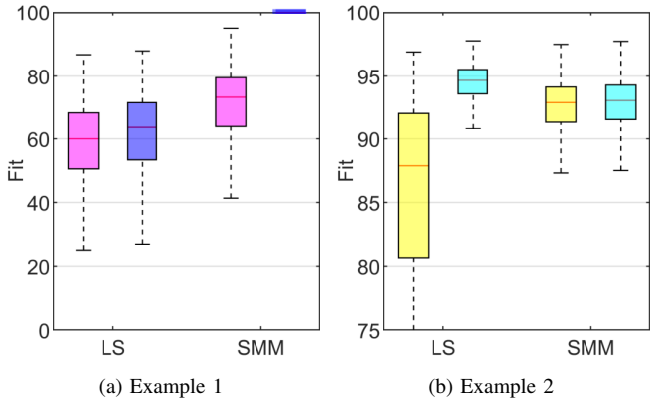


Fig. 3. Comparison of impulse response estimation with unknown input history. Colored area: estimates within two standard deviations.

with impulse response estimates. The boxplots of model fitting for both examples are plotted in Figure 4. For comparison, the case with known input history is also plotted for example 2. The *SMM* algorithm performs better than the *LS* algorithm when the truncation error is large or the input history is unknown. In example 1, the *LS* model fitting is similar for the noisy and noise-free cases, which indicates that the truncation error is the main source of error here. However, when both assumptions of the least squares are satisfied, *LS* performs slightly better than *SMM*. This is due to the fact that part of the data is used to estimate the initial condition in Algorithm 2, whereas it is known for the *LS* algorithm.



(a) Example 1 (b) Example 2

Fig. 4. Box plots of model fitting for both examples with 1000 simulations. In (a), magenta: noisy data, blue: noise-free data. In (b), yellow: unknown input history, cyan: known input history.

VII. Data-Driven Predictive Control With the Signal Matrix Model

In this section, the signal matrix model is used as the predictor in receding horizon predictive control. As discussed in Section IV, the predictor in the unregularized DeePC problem (15) becomes ill-conditioned when noise is present. In the following subsections, we will present two existing methods to remedy this problem, and compare the control performance with the optimal predictor proposed in this paper.

A. Pseudoinverse and Regularized Algorithms

There are mainly two types of existing algorithms to extend (15) to the noisy case: the data-driven subspace predictive control and the regularized DeePC algorithm.

The subspace predictive control approach [20] uses the pseudoinverse solution of g in the predictor instead of the underdetermined linear equality constraints as follows

$$\begin{aligned} \underset{\mathbf{u},\mathbf{y}}{\text{minimize}} \quad & J_{\text{ctr}}(\mathbf{u},\mathbf{y}) \\ \text{subject to} \quad & \mathbf{y} = Y_f\, g_{\text{pinv}}(\mathbf{u};\mathbf{u}_{\text{ini}},\mathbf{y}_{\text{ini}}), \mathbf{u}\in\mathcal{U}, \mathbf{y}\in\mathcal{Y}, \end{aligned} \tag{47}$$

where $g_{\text{pinv}}(\cdot)$ is defined in (16). Multiple applications have been studied with similar algorithms (e.g., [33], [34]). However, as discussed in Section IV, $g_{\text{pinv}}(\cdot)$ is not guaranteed to be an effective choice of g for systems with noise.

The regularized DeePC algorithm [11] adds additional regularization terms in the objective in order to regularize both the norm of g and the slack variable needed to satisfy (11b):

$$\begin{aligned} \underset{\mathbf{u},\mathbf{y},g,\hat{\mathbf{y}}_{\text{ini}}}{\text{minimize}} \quad & J_{\text{ctr}}(\mathbf{u},\mathbf{y}) + \lambda_g \left\|g\right\|_p^p + \lambda_y \left\|\hat{\mathbf{y}}_{\text{ini}} - \mathbf{y}_{\text{ini}}\right\|_p^p \\ \text{subject to} \quad & \text{col}\left(U_p, Y_p, U_f, Y_f\right) g = \text{col}\left(\mathbf{u}_{\text{ini}}, \hat{\mathbf{y}}_{\text{ini}}, \mathbf{u}, \mathbf{y}\right), \\ & \mathbf{u}\in\mathcal{U}, \mathbf{y}\in\mathcal{Y}, \end{aligned} \tag{48}$$

where p is usually selected as 1 or 2. This algorithm can be interpreted as an MPC algorithm acting on the implicit parametric model structure (10) and (11), where the objective is a trade-off between the control performance objective J_{ctl} and the parameter estimation objective

$$J_{\text{id,DeePC}} := \lambda \left\|g\right\|_p^p + \left\|\hat{\mathbf{y}}_{\text{ini}} - \mathbf{y}_{\text{ini}}\right\|_p^p,\ \lambda = \lambda_g/\lambda_y. \tag{49}$$

The set of underdetermined model parameters $(g, \hat{\mathbf{y}}_{\text{ini}})$ are then estimated adaptively in the MPC algorithm. The estimated trajectory in this algorithm is not associated with a fixed input-output mapping, but is biased towards those that predict better control performance. This algorithm is also shown to be effective in multiple applications (e.g., [12], [13]). However, tuning of the regularization parameters is a very hard problem. To the best of our knowledge, there is no practical approach proposed to tune λ_g and λ_y beforehand, and unfortunately the control performance is known to be very sensitive to the regularization parameters [12].

Remark 6: The same data compression scheme as discussed in V-A is applicable to these two algorithms as well.

B. An Optimal Tuning-Free Approach

To address the concerns regarding the two existing methods discussed in the previous subsection, we propose a receding horizon predictive control scheme with the signal matrix model as the predictor. This directly leads to

$$\begin{aligned} \underset{\mathbf{u},\mathbf{y}}{\text{minimize}} \quad & J_{\text{ctr}}(\mathbf{u},\mathbf{y}) \\ \text{subject to} \quad & \mathbf{y} = Y_f\, g_{\text{SMM}}(\mathbf{u};\mathbf{u}_{\text{ini}},\mathbf{y}_{\text{ini}}), \mathbf{u}\in\mathcal{U}, \mathbf{y}\in\mathcal{Y}, \end{aligned} \tag{50}$$

where $g_{\text{SMM}}(\cdot)$ is obtained by Algorithm 2. However, unlike the pseudoinverse predictor where $g_{\text{pinv}}(\cdot)$ is linear with respect to $\mathbf{u}$, the maximum likelihood predictor $g_{\text{SMM}}(\cdot)$ involves an iterative algorithm which cannot be expressed as an equality constraint explicitly.

To solve this problem, we notice that the l_2-norm of g does not change much throughout the receding horizon control, and the algorithm is only iterative with respect to $\|g\|_2^2$. So in a receding horizon context, it makes sense to warm-start the optimization problem from the $\|g\|_2^2$-value at the previous time instant. Then, $g_{\text{SMM}}(\cdot)$ can be closely approximated by the solution of (35) after the first iteration, i.e.,

$$g^t(\mathbf{u};\mathbf{u}_{\text{ini}},\mathbf{y}_{\text{ini}},g^{t-1}) = \mathcal{P}(g^{t-1})\,\mathbf{y}_{\text{ini}} + \mathcal{Q}(g^{t-1})\,\tilde{\mathbf{u}}, \tag{51}$$

where, with an abuse of notation, $g^t(\cdot)$ denotes the approximation of $g_{\text{SMM}}(\cdot)$ with one iteration at time instant t. In this way, the SMM predictor is approximated by a linear equality constraint that can be efficiently solved within a quadratic program. Thus, the proposed approach solves the following optimization problem at each time step

$$\begin{aligned} \underset{\mathbf{u},\mathbf{y}}{\text{minimize}} \quad & J_{\text{ctr}}(\mathbf{u},\mathbf{y}) \\ \text{subject to} \quad & g^t = \mathcal{P}(g^{t-1})\,\mathbf{y}_{\text{ini}} + \mathcal{Q}(g^{t-1})\,\tilde{\mathbf{u}}, \\ & \mathbf{y} = Y_f\, g^t, \mathbf{u}\in\mathcal{U}, \mathbf{y}\in\mathcal{Y}. \end{aligned} \tag{52}$$

The parameter estimation part (49) in regularized DeePC has the same form as the maximum likelihood estimator (33) with $p = 2$, which leads to the predictor in (52). However, our proposed method isolates the parameter estimation part from the control performance objective. More importantly, the problem of hyperparameter tuning is avoided by deriving the coefficients statistically, which requires only information about the noise levels of the offline data σ^2 and the online measurements σ_p^2.

C. Numerical Results

In this subsection, we compare the control performance of three receding horizon predictive control algorithms: 1) *Sub-PC*: subspace predictive control (47), 2) *DeePC*: regularized DeePC (48), and 3) *SMM-PC*: predictive control with the signal matrix model (52). In *DeePC*, the algorithm is tested on a nine-point logarithmic grid of λ_g between 10 and 1000. In this example, the control performance is found to not be sensitive to the value of λ_y, so a fixed value of $\lambda_y = 1000$ is used. In *SMM-PC*, the noise level σ^2 is estimated using (39). Assuming the same sensor for offline and online measurements, we select $\sigma^2 = \sigma_p^2$. To benchmark the performance, we also consider the ideal MPC algorithm (denoted by *MPC*), where both the true state-space model and the noise-free state measurements are available. The result of this benchmark algorithm is thus deterministic and gives the best possible control performance with receding horizon predictive control.

In this example, we consider the LTI system (45). Unless otherwise specified, the following parameters are used in the simulation: $N = 200$, $L_0 = 4$, $L' = 11$, $\sigma^2 = \sigma_p^2 = 1$, $Q = R = 1$. No input and output constraints are enforced, i.e., $\mathcal{U} = \mathbb{R}^{L'n_u}$ and $\mathcal{Y} = \mathbb{R}^{L'n_y}$. A square-wave reference trajectory labeled *Ref* in Figure 7(a) is to be tracked. The offline data are generated with unit i.i.d. Gaussian input signals. For each case, 100 Monte Carlo simulations are conducted. In each run, 60 time steps are simulated. The control performance is assessed

by the true stage cost over all time steps, i.e.,

$$J = \sum_{k=0}^{N_c-1} \left(\left\| y_k^0 - r_k \right\|_Q^2 + \left\| u_k \right\|_R^2 \right), \tag{53}$$

where $N_c = 60$ and y_k^0 is the true output at time k.

When comparing the closed-loop performance, the best choices of λ_g in *DeePC* are selected with an oracle for each run as plotted in Figure 5 (green) for different noise levels. It can be seen that, even for the same control task, the optimal value of this hyperparameter is not only sensitive to the noise level but also to the specific realization of the noise. This makes the tuning process difficult in practice. The optimal value of λ_g is used in all the following simulations. On the other hand, the noise level estimator (39) used in *SMM-PC* is very effective in estimating σ^2 as demonstrated in Figure 5 (yellow).

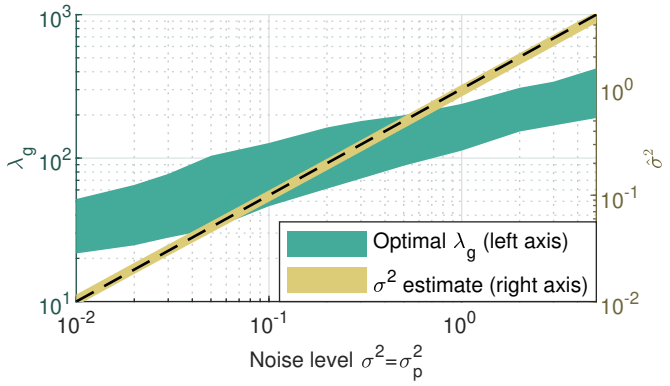


Fig. 5. Hyperparameter tuning in *DeePC* (λ_g) and *SMM-PC* (σ^2) for different noise levels. Colored area: values within one standard deviation. The dashed line shows the true noise level.

The optimization problems are all formulated as quadratic programming problems and solved by MOSEK. The computation time for all three algorithms is similar. The effect of the proposed data compression scheme in Section V-A is illustrated in Figure 6 with the example of *SMM-PC*. By applying the preconditioning, the online computational complexity no longer depends on the data size N.

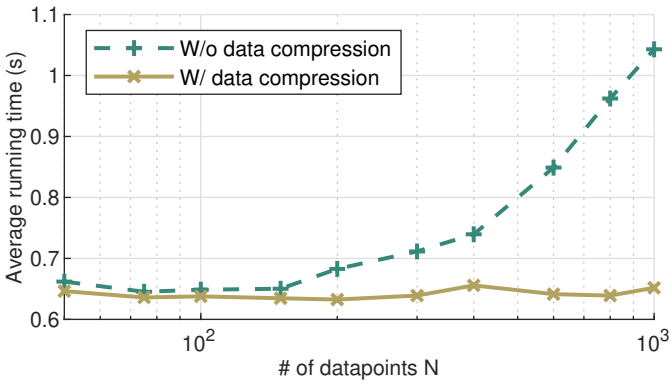


Fig. 6. Average computation time of *SMM-PC* with and without the data compression scheme.

The closed-loop input-output trajectories of different control algorithms are plotted in Figure 7. The closed-loop trajectories are characterized by the range within one standard deviation of the Monte-Carlo simulation. The *SMM-PC* algorithm obtains the closest match to the benchmark trajectory *MPC*. *Sub-PC* applies more aggressive control inputs which results in much larger input costs, whereas the control strategy in *DeePC* is more conservative which results in larger tracking errors. *SMM-PC* also has the smallest variance of input trajectories against different noise realizations. The boxplot of the control performance measure J is shown in Figure 8, which confirms that *SMM-PC* performs better than *Sub-PC* and *DeePC* in this control task, even when the optimal tuning of λ_g, which is not realistic in practice, is used.

The effects of different offline data sizes N and noise levels σ^2, σ_p^2 are investigated in Figure 9. As shown in Figure 9(a), the control performance of *SMM-PC* is not sensitive to the number of datapoints and performs uniformly better among the three algorithms. In fact, good performance is already obtained at only $N = 75$. *DeePC* does not perform very well with small data sizes but gets a similar performance to *SMM-PC* for large N. *Sub-PC* cannot achieve a satisfying result even with a large data size because, as discussed in Section V-C, the subspace predictor is problematic to deal with online measurement noise σ_p^2, which cannot be averaged out by a large N. Figure 9(b) shows that all three algorithms perform similarly at low noise levels as they are all stochastic variants of the noise-free algorithm (15). *SMM-PC* obtains slightly worse results under low noise levels ($\sigma^2 = \sigma_p^2 < 0.05$) compared to the optimally tuned *DeePC* with an oracle, but the performance improvement of *SMM-PC* is significant for higher noise levels.

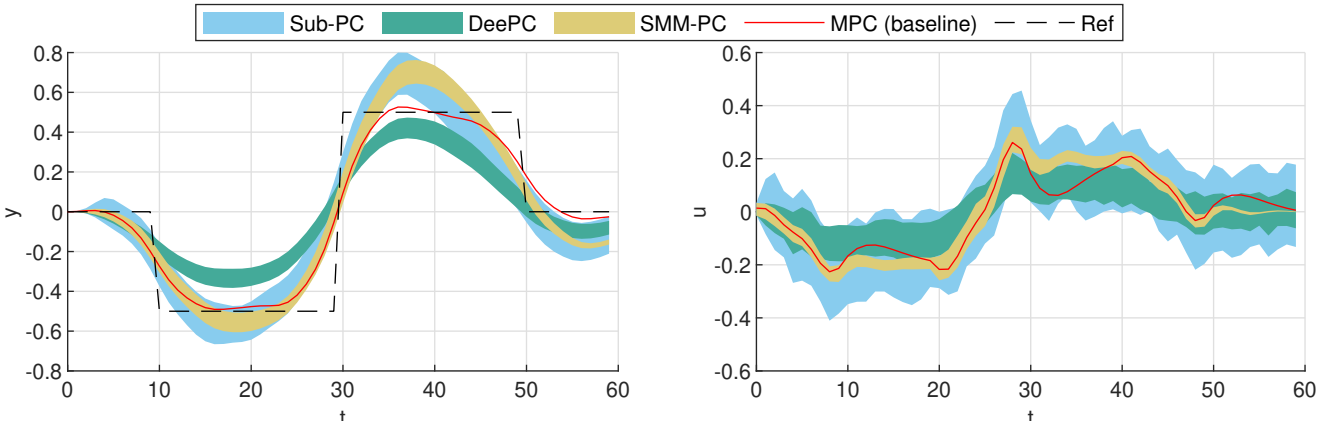
VIII. Conclusions

In this work, we propose a novel statistical framework to estimate data-driven models from large noise-corrupted datasets. This is formulated as a maximum likelihood estimation problem. The problem is solved efficiently by approximating it as a sequential quadratic program with data compression and data-driven noise level estimation. This framework extends the current works on data-driven methods to noisy data by providing an optimal solution to the underdetermined implicit model structure and establishing the signal matrix model.

With the signal matrix model, two approaches in system identification and receding horizon control are developed. They obtain an impulse response estimate with less restrictive assumptions and an effective tuning-free data-driven receding horizon control algorithm respectively. The results from these two approaches demonstrate that the proposed framework can improve the state-of-the-art methods in both data-driven simulation and control in the presence of noisy data.

Appendix I
Derivation of Σ_y

Let $\zeta_i \in \mathbb{R}^L$, $i = 1, \ldots, LM$ be the i-th column of $(g^\mathsf{T} \otimes \mathbb{I})$, $\mathcal{S} = \left\{ (i,j) \middle| (\mathrm{vec}(Y))_i = (\mathrm{vec}(Y))_j \right\}$, and $\Sigma_{y1} =$



(a) Output trajectory (b) Input trajectory

Fig. 7. Comparison of closed-loop input-output trajectories with different control algorithms. Colored area: trajectories within one standard deviation.

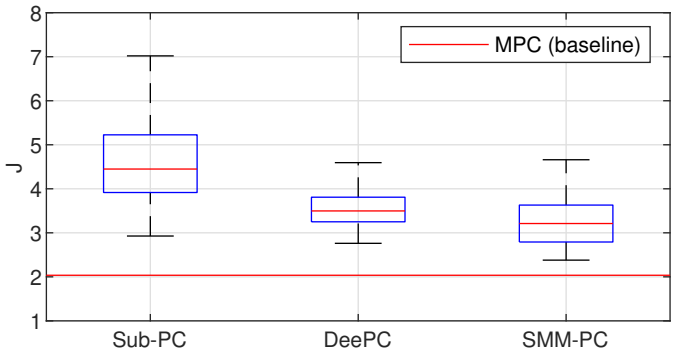
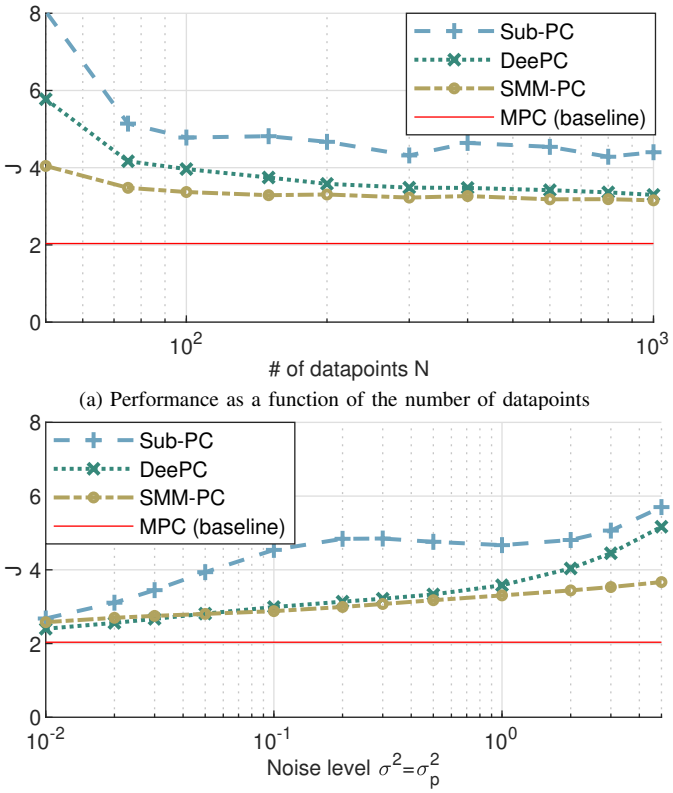


Fig. 8. Comparison of the control performance in terms of total stage costs J with different control algorithms with 100 simulations ($\boldsymbol{\sigma^2 = \sigma_p^2 = 1, N = 200}$).



(a) Performance as a function of the number of datapoints

(b) Performance as a function of the noise level

Fig. 9. The effect of different offline data sizes and noise levels on the control performance.

$\left(g^\mathsf{T} \otimes \mathbb{I}\right) \Sigma_{yd} \left(g \otimes \mathbb{I}\right)$. According to (24), we have

$$\Sigma_{y1} = \sigma^2 \sum_{(i,j)\in\mathcal{S}} \zeta_i \zeta_j^\mathsf{T}. \tag{54}$$

Let the i-th and the j-th entries of $\mathrm{vec}(Y)$ correspond to the (q,r)-th and the (s,t)-th entries of Y respectively, i.e., $i = (r-1)L+q$, $j = (t-1)L+s$. From the Hankel structure, the pair $(i,j) \in \mathcal{S}$ iff $q+r = s+t$. According to the structure of $\left(g^\mathsf{T} \otimes \mathbb{I}\right)$, we have $\zeta_i = g_r \mathbf{e}_q$, $\zeta_j = g_t \mathbf{e}_s$, where $\mathbf{e}_q \in \mathbb{R}^L$ is the unit vector with q-th non-zero entry, and similarly for $\mathbf{e}_s$. Thus,

$$\Sigma_{y1} = \sigma^2 \sum_{q+r=s+t} g_r g_t \mathbf{e}_q \mathbf{e}_s^\mathsf{T}. \tag{55}$$

So the (q,s)-th entry of Σ_{y1} is given by

$$\left(\Sigma_{y1}\right)_{q,s} = \sigma^2 \sum_{q+r=s+t} g_r g_t, \tag{56}$$

which directly leads to (25).

Appendix II
Proof of Proposition 2

Let $K_\lambda = F^{-1} - F^{-1}U^\mathsf{T}(UF^{-1}U^\mathsf{T})^{-1}UF^{-1}$ and $g_\lambda = K_\lambda Y_p^\mathsf{T} \mathbf{y}_{\text{ini}} + \mathcal{Q}\tilde{\mathbf{u}}$. From the structure of (33), when $\lambda \to 0$, g_λ converges to g_{pinv}. When $\lambda = L'\sigma_p^2 / \left\|g_\lambda\right\|_2^2$, $g_\lambda = g_{\text{SMM}}$. Then we have $\mathrm{cov}(g_\lambda) = \sigma_p^2 (K_\lambda Y_p^\mathsf{T})(K_\lambda Y_p^\mathsf{T})^\mathsf{T}$. The derivative of $\mathrm{tr}(\mathrm{cov}(g_\lambda))$ with respect to λ is calculated as follows.

$$\begin{aligned}\frac{\partial \,\mathrm{tr}(\mathrm{cov}(g_\lambda))}{\partial \left(F^{-1}\right)_{i,j}} &= \mathrm{tr}\left[\left(\frac{\partial \,\mathrm{tr}(\mathrm{cov}(g_\lambda))}{\partial K_\lambda}\right)^\mathsf{T} \frac{\partial K_\lambda}{\partial \left(F^{-1}\right)_{i,j}}\right] \\ &= 2\sigma_p^2 \,\mathrm{tr}\left[\left(Y_p^\mathsf{T} Y_p K_\lambda\right)^\mathsf{T} K_\lambda F \Delta(i,j) F K_\lambda\right],\end{aligned} \tag{57}$$

where the (i,j)-th element of $\Delta(i,j) \in \mathbb{R}^{M\times M}$ is 1 and the other elements are 0. Then,

$$\begin{aligned}\frac{\partial \operatorname{tr}(\operatorname{cov}(g_\lambda))}{\partial \lambda} &= \operatorname{tr}\left[\left(\frac{\partial \operatorname{tr}(\operatorname{cov}(g_\lambda))}{\partial F^{-1}}\right)^{\mathsf{T}} \frac{\partial F^{-1}}{\partial \lambda}\right] \\ &= -2\sigma_p^2 \operatorname{tr}\left[\left(FK_\lambda \left(Y_p^{\mathsf{T}} Y_p K_\lambda\right)^{\mathsf{T}} K_\lambda F\right)^{\mathsf{T}} F^{-2}\right] \\ &= -2\sigma_p^2 \operatorname{tr}\left(K_\lambda Y_p^{\mathsf{T}} Y_p K_\lambda K_\lambda\right), \end{aligned} \tag{58}$$

According to the Schur complement, since

$$\begin{bmatrix} F^{-1} & F^{-1}U^{\mathsf{T}} \\ UF^{-1} & UF^{-1}U^{\mathsf{T}} \end{bmatrix} = \begin{bmatrix} \mathbb{I} \\ U \end{bmatrix} F^{-1} \begin{bmatrix} \mathbb{I} & U^T \end{bmatrix} \succ 0, \tag{59}$$

we have $K_\lambda \succ 0$. Together with $K_\lambda Y_p^{\mathsf{T}} Y_p K_\lambda \succ 0$, we have $\partial \operatorname{tr}(\operatorname{cov}(g_\lambda))/\partial\lambda < 0$ for all λ. This directly leads to Proposition 2.

REFERENCES

[1] 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.

[2] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[3] H. Hjalmarsson, "From experiment design to closed-loop control," *Automatica*, vol. 41, no. 3, pp. 393–438, 2005.

[4] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 253–279, 2019.

[5] H. J. van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4753–4768, 2020.

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

[7] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[8] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[9] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi, "Willems' fundamental lemma for state-space systems and its extension to multiple datasets," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 602–607, 2020.

[10] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*. Cham, Switzerland: Springer, 2016.

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

[12] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for grid-connected power converters," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 8130–8135.

[13] J. Coulson, J. Lygeros, and F. Dörfler, "Regularized and distributionally robust data-enabled predictive control," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 2696–2701.

[14] D. Alpago, F. Dörfler, and J. Lygeros, "An extended Kalman filter for data-enabled predictive control," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 994–999, 2020.

[15] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2021.

[16] M. Yin, A. Iannelli, and R. S. Smith, "Maximum likelihood signal matrix model for data-driven predictive control," in *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, vol. 144, 2021, pp. 1004–1014.

[17] M. Moonen, B. D. Moor, L. Vandenberghe, and J. Vandewalle, "On- and off-line identification of linear state-space models," *International Journal of Control*, vol. 49, no. 1, pp. 219–232, 1989.

[18] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[19] W. Favoreel, B. D. Moor, and M. Gevers, "SPC: Subspace predictive control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004–4009, 1999.

[20] S. Sedghizadeh and S. Beheshti, "Data-driven subspace predictive control: Stability and horizon tuning," *Journal of the Franklin Institute*, vol. 355, no. 15, pp. 7509–7547, 2018.

[21] I. Markovsky, J. C. Willems, P. Rapisarda, and B. L. D. Moor, "Data driven simulation with application to system identification," in *16th IFAC World Congress*, vol. 38, no. 1. Elsevier BV, 2005, pp. 970–975.

[22] G. Q. Carapia, I. Markovsky, R. Pintelon, P. Z. Csurcsia, and D. Verbeke, "Experimental validation of a data-driven step input estimation method for dynamic measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4843–4851, 2020.

[23] H. J. van Waarde, M. K. Camlibel, and M. Mesbahi, "From noisy data to feedback controllers: non-conservative design via a matrix S-lemma," *arXiv preprint arXiv:2006.00870*, 2020.

[24] J. C. Willems and J. W. Polderman, *Introduction to Mathematical Systems Theory: A Behavioral Approach*. New York, NY, USA: Springer, 1997, vol. 26.

[25] I. Markovsky and F. Dörfler, "Identifiability in the behavioral setting," 2020, submitted for publication. Available: http://homepages.vub.ac.be/˜imarkovs/publications/identifiability.pdf.

[26] Y. Yu, S. Talebi, H. J. van Waarde, U. Topcu, M. Mesbahi, and B. Açıkmeşe, "On controllability and persistency of excitation in data-driven control: Extensions of willems' fundamental lemma," *arXiv preprint arXiv:2102.02953*, 2021.

[27] I. Markovsky and S. V. Huffel, "Overview of total least-squares methods," *Signal Processing*, vol. 87, no. 10, pp. 2283–2302, 2007.

[28] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, no. 1, pp. 1–51, 1995.

[29] A. Damen, P. Van den Hof, and A. Hajdasinski, "Approximate realization based upon an alternative to the Hankel matrix: the Page matrix," *Systems & Control Letters*, vol. 2, no. 4, pp. 202–208, 1982.

[30] A. Iannelli, M. Yin, and R. S. Smith, "Design of input for data-driven simulation with Hankel and Page matrices," *arXiv preprint arXiv:2109.04742*, 2021.

[31] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, 2014.

[32] G. Pillonetto and G. D. Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol. 46, no. 1, pp. 81–93, 2010.

[33] R. Hallouzi and M. Verhaegen, "Fault-tolerant subspace predictive control applied to a Boeing 747 model," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 873–883, 2008.

[34] R. Kadali, B. Huang, and A. Rossiter, "A data driven subspace approach to predictive controller design," *Control Engineering Practice*, vol. 11, no. 3, pp. 261–278, 2003.