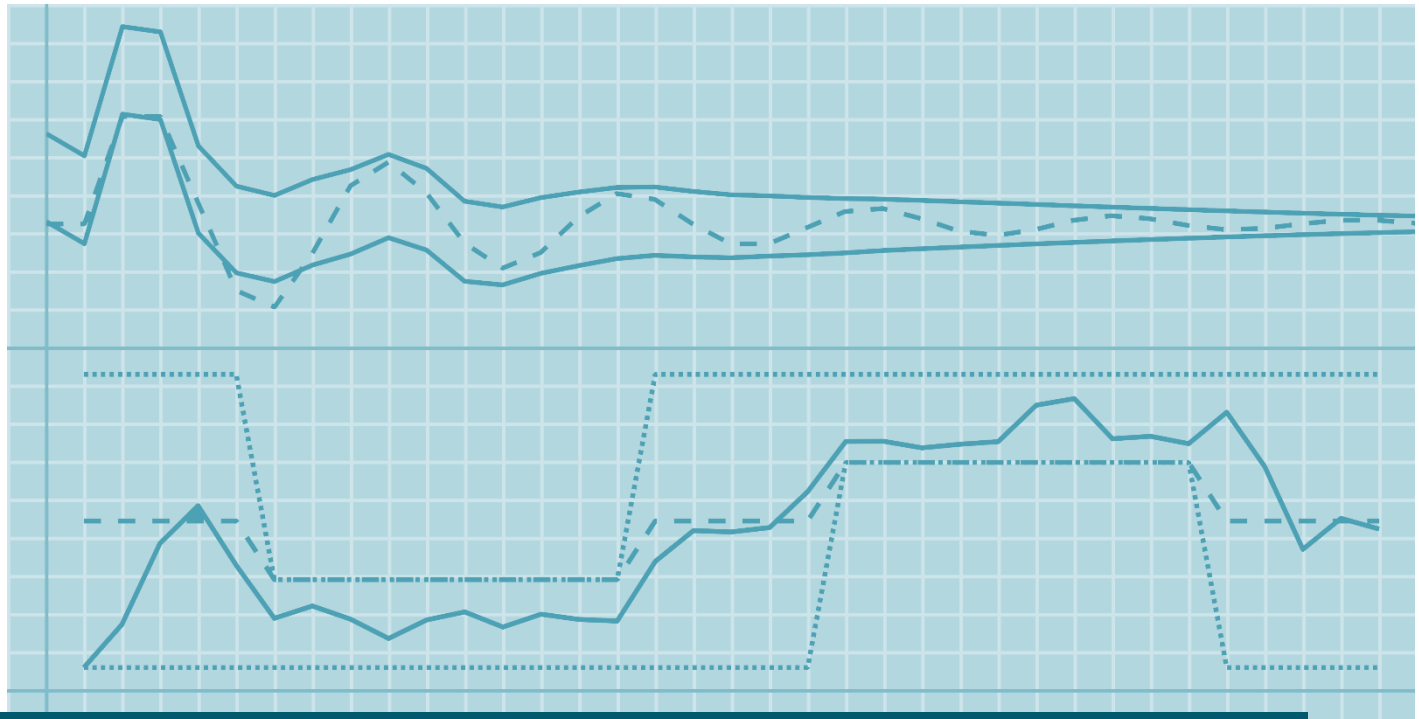
ETH zürich

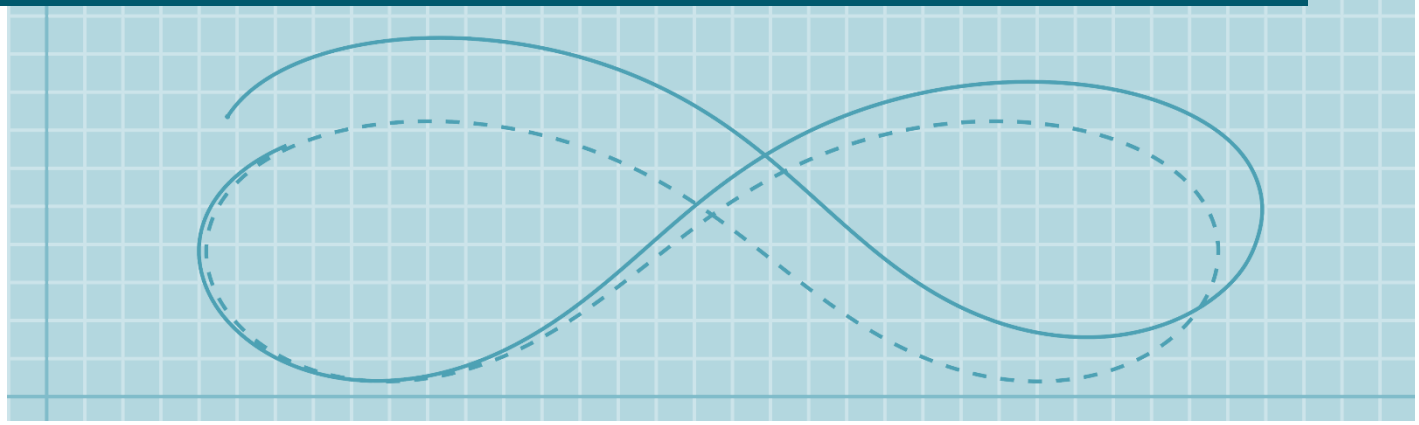
AUTOMATIC CONTROL LABORATORY



# Regularized and Nonparametric Approaches in System Identification and Data-Driven Control

Mingzhou Yin | 殷明周

Diss. ETH No. 30027



DITET

DISS. ETH NO. 30027

# REGULARIZED AND NONPARAMETRIC APPROACHES IN SYSTEM IDENTIFICATION AND DATA-DRIVEN CONTROL

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES
(Dr. sc. ETH Zurich)

presented by

MINGZHOU YIN

MSc, Delft University of Technology

born on 29.03.1994

accepted on the recommendation of

Prof. Dr. Roy S. Smith, examiner
Prof. Dr. Florian Dörfler, co-examiner
Prof. Dr. Alessandro Chiuso, co-examiner

2024

ETH Zurich
Automatic Control Laboratory
Physikstrasse 3
8092 Zurich, Switzerland

*To my family*

# Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor, Prof. Roy S. Smith, for providing me with the incredible opportunity to work with him in the Automatic Control Laboratory (IfA) at ETH Zurich over the past five years. I am profoundly grateful for the freedom, guidance, and support he generously offered throughout my academic journey. His extensive knowledge and expertise in control theory have been invaluable assets to my doctoral studies.

I want to thank my second advisor and co-examiner, Prof. Florian Dörfler, for agreeing to serve in this capacity and for introducing me to the world of data-driven control with his inspiring works. I am also thankful to my co-examiner, Prof. Alessandro Chiuso, for agreeing to join the committee and assess my thesis.

Heartfelt thanks are extended to my collaborators who played a significant role in my research as well as my development as a researcher. I am indebted to Andrea Iannelli for his insightful discussions, encouragement, and continuous support over the years. His influence not only permeates the majority of my research output but also profoundly impacts my understanding of what it means to be an exceptional colleague and researcher. I appreciate Mohammad Khosravi for immersing me in the intricacies of the kernel method and for providing invaluable mathematical insights. Acknowledgment goes to Hanmin Cai and other colleagues at Empa for making it possible to apply my work to real-world scenarios. I am also thankful to Amber Srivastava, Mohamed Abdalmoaty, and Jared Miller for our enriching collaborations, as well as to everyone in our group for sharing interesting research ideas and constructive comments in and beyond the group meetings.

I am thankful to the students I had the privilege of working with—Defne Ege Ozan, Mehmet Tolga Akan, Andrea Gattiglio, Kailai Yan, Ran Chen, David Laseca Pérez, and Yanze Liu—for their commitment and hard work, as well as my co-supervisors. They not only played an integral role in advancing my research but also significantly contributed to my personal growth in project management. I learned a lot from each of them.

I would like to express my appreciation to all the members of the lab for fostering a positive and friendly environment. I would like to thank the teaching assistant team of the system identification course—Mathias Hudoba de Badyn, Anil Parsi, Pier Giuseppe Sessa, Francesco Micheli, Samuel Balula, Amber Srivastava, and Aren Karapetyan—for their support and hard work, especially

before exam weeks. My thanks go to Anil Parsi for organizing the enjoyable group dinners. I am thankful to Ezzat Elokda and Varsha Behrunani for making the IfA Open House a success together. Thank you Sabrina Baumann and Tanja Turner for maintaining everything smooth and comfortable in the lab.

Finally, I express my deepest gratitude to my wife, Xiao Tan, who has been the best partner throughout this journey. Her love, support, and laughter have been indispensable, and I could not have completed this academic pursuit without her. Special thanks also go to my family for their unwavering support of my decisions.

*Zurich, February 19, 2024* Mingzhou Yin

# Abstract

This thesis delves into regularized and nonparametric approaches in system identification and data-driven control. Classical model-based control design relies on a compact parametric model structure, which is difficult to obtain for modern complex systems. To address this challenge, regularized approaches adopt general high-dimensional model structures and apply sparse learning and kernel learning theories to identify models by leveraging the sparsity and smoothness properties of the system, respectively. In sparse learning, atomic norm regularization is employed to learn the sparse pole locations of the system within the unit disk. A novel algorithm is presented to solve the associated infinite-dimensional sparse learning problem. Debiasing and stability selection algorithms are applied to enhance the identification performance as well. In kernel learning, a multiple kernel design with optimal first-order kernels is proposed to identify the impulse response of the system. This enforces a low-complexity model structure while maintaining the favorable bias-variance trade-off property of kernel learning. More reliable error bounds, associated with the Gaussian process interpretation of kernel learning, are derived when hyperparameters are unknown, supporting safety-critical applications.

An alternative path to circumvent model structure selection is to construct nonparametric predictors that predict output trajectories. This can be done by characterizing possible system behaviors as linear combinations of deterministic trajectory data. Extensions of this approach to stochastic data are investigated. A novel algorithm is developed to denoise the data by solving a low-rank Hankel matrix denoising problem. It achieves a more substantial noise reduction than existing algorithms. A maximum likelihood predictor, dubbed the signal matrix model, is derived to establish a statistical framework that provides accurate prediction in the presence of noise without requiring sophisticated tuning. Prediction error quantification associated with the nominal prediction is also provided. The proposed predictor can be directly applied to receding horizon predictive control, replacing model-based predictors, with the possibility to incorporate online data. It demonstrates superior performance compared to existing data-driven predictors. The algorithm is further extended to the stochastic control framework with initial condition estimation and guaranteed constraint satisfaction. Its effectiveness in practice is validated through high-fidelity simulation of a space heating control case study.

Specific identification approaches for periodic systems are also studied. Linear time-periodic systems are identified by reformulating them into switched systems and extending the atomic

norm regularization approach with grouped variables. In the frequency domain, a novel subspace identification algorithm is proposed by estimating the time-aliased periodic impulse response from the frequency response of the lifted system. Periodic models can also be utilized to identify local limit cycle dynamics. This is accomplished by linearizing the system along the limit cycle and estimating the periodic dynamics matrix of the linearized system by kernel learning. The approach is tested on an airborne wind energy system.

# Zusammenfassung

Diese Arbeit befasst sich mit regulierten und nichtparametrischen Ansätzen in der Systemidentifikation und datengesteuerten Steuerung. Klassisches, modellbasiertes Steuerungsdesign stützt sich auf eine kompakte parametrische Modellstruktur, die jedoch für moderne komplexe Systeme schwer zu erhalten ist. Um diese Herausforderung zu bewältigen, adoptieren regulierte Ansätze allgemeine hochdimensionale Modellstrukturen und wenden sparsames Lernen und Kernel-Lernen an, um Modelle zu identifizieren, indem sie die Sparsamkeit und Glätteeigenschaften des Systems nutzen. Beim sparsamen Lernen wird die atomare Normregularisierung verwendet, um die spärlichen Polpositionen des Systems innerhalb der Einheitsdisk zu lernen. Ein neuartiger Algorithmus wird präsentiert, um das damit verbundene unendlich-dimensionale sparsame Lernproblem zu lösen. Entzerrungs- und Stabilitätsauswahlalgorithmen werden ebenfalls angewandt, um die Identifikationsleistung zu verbessern. Im Kernel-Lernen wird ein multiples Kernel-Design mit optimalen erststufigen Kernels vorgeschlagen, um die Impulsantwort des Systems zu identifizieren. Dies erzwingt eine niedrigkomplexe Modellstruktur und behält dabei die günstige Eigenschaft des Bias-Varianz-Ausgleichs des Kernel-Lernens bei. Zuverlässigere Fehlergrenzen, die mit der Gaußschen Prozessinterpretation des Kernel-Lernens verbunden sind, werden abgeleitet, wenn die Hyperparameter unbekannt sind, was sicherheitskritische Anwendungen unterstützt.

Ein alternativer Weg, die Auswahl der Modellstruktur zu umgehen, besteht darin, nichtparametrische Prädiktoren zu konstruieren, die Ausgabetrajektorien vorhersagen. Dies kann geschehen, indem mögliche Systemverhaltensweisen als lineare Kombinationen von deterministischen Trajektoriendaten charakterisiert werden. Erweiterungen dieses Ansatzes für stochastische Daten werden untersucht. Ein neuartiger Algorithmus wird entwickelt, um die Daten durch Lösen eines Rangminderungsproblems der Hankel-Matrix zu entrauschen. Dies erreicht eine wesentlich stärkere Geräuschreduzierung als bestehende Algorithmen. Ein Maximum-Likelihood-Prädiktor, genannt das Signal-Matrix-Modell, wird abgeleitet, um einen statistischen Rahmen zu schaffen, der genaue Vorhersagen in Gegenwart von Lärm ohne ausgefeilte Abstimmung ermöglicht. Eine Quantifizierung des Vorhersagefehlers in Verbindung mit der nominalen Vorhersage wird ebenfalls bereitgestellt. Der vorgeschlagene Prädiktor kann direkt auf die rückwärtige Horizont-Vorhersagesteuerung angewendet werden und ersetzt modellbasierte Prädiktoren mit der Möglichkeit, Online-Daten einzubeziehen. Er zeigt eine überlegene Leistung im Vergleich zu bestehenden datengesteuerten Prädiktoren. Der Algorithmus wird weiter auf das stochastische Steuerungsfra-

mework mit anfänglicher Zustandsschätzung und garantiertem Restriktionserfüllung ausgedehnt. Seine Wirksamkeit in der Praxis wird durch hochtreue Simulation eines Raumheizungssteuerung-Fallbeispiels validiert.

Spezifische Identifikationsansätze für periodische Systeme werden ebenfalls untersucht. Lineare zeitperiodische Systeme werden identifiziert, indem sie in geschaltete Systeme umformuliert und der atomaren Normregularisierungsansatz mit gruppierten Variablen erweitert wird. Im Frequenzbereich wird ein neuartiger Subraum-Identifikationsalgorithmus vorgeschlagen, indem die zeitlich verzerrte periodische Impulsantwort aus der Frequenzantwort des gehobenen Systems geschätzt wird. Periodische Modelle können auch genutzt werden, um lokale Grenzzyklusdynamiken zu identifizieren. Dies wird erreicht, indem das System entlang des Grenzzyklus linearisiert und die periodische Dynamikmatrix des linearisierten Systems durch Kernel-Lernen geschätzt wird. Der Ansatz wird an einem luftgebundenen Windenergiesystem getestet.

# Abbreviations

| | |
|---|---|
| **ARX** | autoregressive with extra input |
| **CPSS** | complementary pairs stability selection |
| **DCP** | difference of convex programming |
| **DDPC** | data-driven predictive control |
| **DeePC** | data-enabled predictive control |
| **DFT** | discrete Fourier transform |
| **DI** | diagonal |
| **ETFE** | empirical transfer function estimation |
| **EYM** | Eckart-Young-Mirsky |
| **FIR** | finite impulse response |
| **GP** | Gaussian process |
| **HTF** | harmonic transfer function |
| **i.i.d.** | independent and identically distributed |
| **IDFT** | inverse discrete Fourier transform |
| **IIR** | infinite impulse response |
| **lasso** | least absolute shrinkage and selection operator |
| **LPPV** | linear periodically parameter-varying |
| **LPV** | linear parameter-varying |
| **LTI** | linear time-invariant |
| **LTP** | linear time-periodic |
| **LTV** | linear time-varying |
| **MAP** | maximum *a posteriori* |
| **MIMO** | multiple-input multiple-output |
| **MISO** | multiple-input single-output |
| **MLE** | maximum likelihood estimation |
| **MPC** | model predictive control |
| **MSE** | mean squared error |
| **ODE** | ordinary differential equation |

| | |
|---|---|
| **PEM** | prediction error method |
| **QP** | quadratic programming |
| **RKHS** | reproducing kernel Hilbert space |
| **SDP** | semidefinite programming |
| **SE** | squared exponential |
| **SISO** | single-input single-output |
| **SLRA** | structured low-rank approximation |
| **SMM** | signal matrix model |
| **SMM-PC** | signal matrix model predictive control |
| **SNR** | signal-to-noise ratio |
| **SOC** | second-order cone |
| **SPC** | subspace predictive control |
| **SQP** | sequential quadratic programming |
| **SS** | stable spline |
| **SVD** | singular value decomposition |
| **SVM** | support vector machine |
| **TC** | tuned/correlated |
| **TSVD** | truncated singular value decomposition |
| **UMAR** | urban mining and recycling |
| **WD** | Wasserstein distance |
| **WFL** | Willems' fundamental lemma |
| **w.p.** | with probability |

# Nomenclature

| | |
|---|---|
| $\#[\cdot]$ | the number of elements in a set |
| $\mathrm{col}\,(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ | row-wise concatenation $[\mathbf{x}_1^\top \ \ldots \ \mathbf{x}_n^\top]^\top$ |
| $\Im(\cdot)$ | the imaginary part of a complex number |
| $\lfloor \cdot \rfloor$ | the floor function |
| $\mathbb{C}$ | the set of complex numbers |
| $\mathbb{C}^n$ | the $n$-dimensional complex space |
| $\mathbb{E}(\cdot)$ | the expected value |
| $\mathbb{I}$ | the identity matrix |
| $\mathbb{P}(\cdot)$ | the probability of a random event |
| $\mathbb{R}$ | the set of real numbers |
| $\mathbb{R}^n$ | the $n$-dimensional Euclidean space |
| $\mathbb{R}_+$ | the set of non-negative real numbers |
| $\mathbb{R}_{++}$ | the set of positive real numbers |
| $\mathbb{S}^n_{++}$ | the set of $n$-by-$n$ positive definite matrices |
| $\mathbb{S}^n_{+}$ | the set of $n$-by-$n$ positive semi-definite matrices |
| $\mathbb{Z}$ | the set of integers |
| $\mathbb{Z}_+$ | the set of non-negative integers |
| $\mathbf{0}$ | a vector of zeros |
| $\mathbf{1}$ | a vector of ones |
| $\|\cdot\|_*$ | the nuclear norm of a matrix |
| $\|\cdot\|_F$ | the Frobenius norm of a matrix |
| $\|x\|_P$ | the weighted $l_2$-norm $\left(x^\top P x\right)^{\frac{1}{2}}$ |
| $\otimes$ | the Kronecker product |
| $\overset{p}{\leq}$ | less than or equal to with probability $p$ |
| $\partial f$ | the subdifferential of $f$ |
| $\mathcal{N}(\mu, \sigma^2)$ | Gaussian distribution with a mean of $\mu$ and a variance of $\sigma^2$ |
| $\Re(\cdot)$ | the real part of a complex number |

| | |
|---|---|
| $\rho(\cdot)$ | the spectral radius of a matrix |
| $\mathrm{blkdiag}(\cdot)$ | diagonal-wise concatenation of matrices |
| $\mathrm{cov}(\cdot)$ | the covariance |
| $\mathrm{diag}(\cdot)$ | the vector of diagonal elements of a square matrix |
| $\mathrm{range}(\cdot)$ | the range (column space) of a matrix |
| $\mathrm{rank}(\cdot)$ | the rank of a matrix |
| $\mathrm{span}(\cdot)$ | the linear span of a set of vectors |
| $\mathrm{std}(\cdot)$ | the standard deviation |
| $\mathrm{tr}(\cdot)$ | the trace of a matrix |
| $\mathrm{vec}(\cdot)$ | the vectorization operator by stacking columns into one vector |
| $\mathbb{0}$ | a matrix of zeros |
| $A \preccurlyeq B$ | $(B-A)$ is positive semidefinite |
| $A^{\mathsf{H}}$ | the Hermitian (conjugate transpose) of matrix $A$ |
| $A^{\top}$ | the transpose of matrix $A$ |
| $f(x) = O(g(x))$ | there exists $M > 0$, $x_0$, such that $\lvert f(x)\rvert \leq Mg(x)$ for all $x \geq x_0$ |
| $j$ | the imaginary unit |
| $X^{\dagger}$ | the Moore-Penrose pseudoinverse of $X$ |

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Automatic control is a ubiquitous technology supporting the intelligent and autonomous operation of various dynamical systems. It provides autonomous algorithms to achieve desired performance objectives by manipulating controllable inputs to the system based on knowledge and measurements of the system. Traditionally, such knowledge is encoded in a compact mathematical form known as a model that describes the system's behaviors. Various control design approaches have been proposed, analyzed, and applied under the premise that such a model is accessible for predicting the responses of the system.

## 1.1 The Paradigm of System Identification

The step of estimating a model of a dynamical system from observed data is known as system identification. A classical system identification procedure includes the following four steps.

1. *Data collection*: Trajectories of the system inputs and responses are recorded. Such trajectories can come from specifically designed identification experiments or historical data by running the actual system or a high-fidelity simulation model. Such data are known as *identification data*.
2. *Selection of model structure*: For small-scale physical systems, parametric models can often be obtained from first principles with a small number of parameters to be identified. Such models are known as "grey-box" models. For more complex systems, first-principle models may not be available. In this case, general low-dimensional model structures, known as "black-box" models, are used. Parameters in model structure selections are known as *hyperparameters*.
3. *Determining the "best" model*: The "best" model is often selected to minimize the difference between collected and predicted system responses. This is often set up as regression problems.
4. *Model validation*: The best hyperparameters are selected by testing models on new system trajectory data collected independently of the identification data or using information

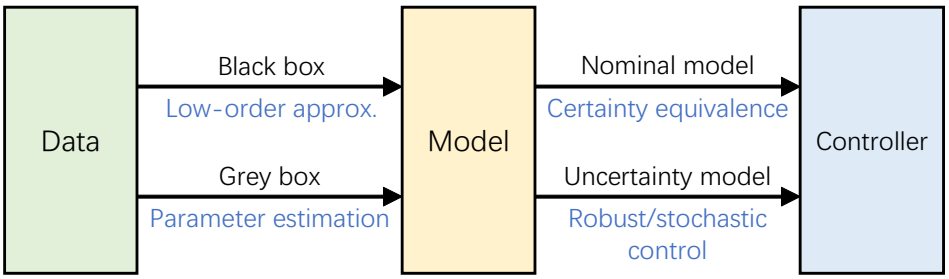



Figure 1.1: Paradigm of system identification.

criteria.

After identifying a model, control design is often conducted by assuming that the model fully captures the true system behaviors. This assumption is known as the *certainty equivalence* principle. Alternatively, an uncertainty model can be identified, which describes how the true system behaviors can differ from those predicted by the identified model. Control design approaches considering such uncertainties are known as robust or stochastic control.

In this way, the control design problem is split into two distinctive steps via mathematical abstraction using models, namely system identification and model-based control design. Such approaches are known as the paradigm of system identification and are summarized in Figure 1.1. In recent years, this framework of using data in control design is also known as indirect data-driven control (Dörfler et al., 2023).

Classical algorithms in system identification approach the problem in a parameter estimation framework, where a finite-dimensional parametrization models the system. Tools in classical statistics can then estimate the optimal model parameters. One well-known approach in this category is the prediction error method (PEM) based on maximum likelihood estimation (MLE) (Åström, 1980). Despite some numerical difficulties, this framework has succeeded in various applications (Ljung, 1999). However, an essential requirement for this framework to function is that suitable model structure and effective model complexity selection are available.

## 1.2 System Identification as Learning Problem

One insight that has been widely leveraged in recent studies of system identification is that it shares many similarities with supervised learning, despite that different terminology is used, e.g., training vs. identification, algorithm vs. estimator, and overfitting vs. bias-variance trade-off.

One main paradigm shift resulting from the learning perspective of system identification is the use of nonparametric and high-dimensional statistics in system identification. Let $n$ be the number of parameters in the model, and $N$ be the number of data points. Classical identification methods work in the regime where $n \ll N$, whereas modern approaches tend to work with higher

Table 1.1: Paradigm shift in system identification.

| Method | Parameter estimation | Kernel learning | Sparse learning |
|---|---|---|---|
| **Theory** | Classical stat. | Nonparametric stat. | High-dimensional stat. |
| **Regime** | $n \ll N$ | $n \approx N$ | $n \gg N$ |
| **Prior info.** | Low dimension | Smoothness | Sparsity |
| **Tool** | MLE | RKHS, GP | Lasso, compressive sensing |
| **Algorithm** | PEM | Kernel-based identification | Atomic norm regularization |
| **Problem** | Model structure selection | Complexity measure | Bias, false positives |

dimensions. When a nonparametric model ($n \approx N$) is used, kernel learning can be applied using tools including reproducing kernel Hilbert space (RKHS) and Gaussian process (GP). When an over-parametrized model ($n \gg N$) is used, sparse learning comes into use with the help of the least absolute shrinkage and selection operator (lasso) and compressive sensing techniques. The characteristics of the methods are summarized in Table 1.1.

This paradigm shift arises timely as more complex and large-scale systems have emerged recently. The problem of model structure and complexity selection in classical system identification becomes more and more challenging as low-dimensional model structures become less accessible. On the other hand, increasing computational capability makes it possible to work directly with nonparametric and over-parametrized models in model-based control design.

The above backgrounds have led to a surge in learning-based identification methods in recent years. The critical idea in such ways is the introduction of regularization (Ljung et al., 2019; Pillonetto et al., 2016). Instead of just looking for the best adherence to identification data, regularized system identification solves a bi-objective problem, which minimizes a combination of a loss function measuring data fitting and a *regularizer* encoding prior model knowledge in system theory, such as stability, model complexity, frequency domain information (Pillonetto et al., 2014; Chen, 2018; Marconato et al., 2016; Shah et al., 2012; Khosravi, 2021). In this way, prior knowledge is integrated in the regularizer instead of in the model structure as in classical system identification. This significantly improves the estimation quality of high-dimensional models, which would otherwise lead to overfitting.

In Part I of the thesis, we focus on atomic norm regularization algorithms based on sparse learning and kernel-based identification based on kernel learning in Chapter 2 and Chapter 3, respectively.

### 1.2.1 Atomic Norm Regularization

Atomic norm regularization in system identification focuses on simultaneous model order control and model fitting without prior model order selection. It models the system as a summation

of “atoms", which are some predefined basis models. The number of atoms is typically much larger than the data length $N$, leading to the high-dimensional statistics regime. Still, only a sparse selection of them is active in the model. The model complexity can then be controlled by regularizing the $l_1$-norm of the coefficients. This is known as regularizing the atomic norm with respect to the atomic decomposition (Shah et al., 2012). When first-order atoms are selected, this results in a lasso-type problem that promotes models with a small number of poles (Yuan and Lin, 2006). Another advantage of the first-order atomic decomposition is that it directly identifies the pole locations of the system, rather than polynomial coefficients as in conventional models such as autoregressive with extra input (ARX) models. Pole locations are essential in classical control design yet hard to estimate with conventional identification approaches.

However, existing work on the atomic norm regularization approach has multiple known drawbacks. First, instead of solving the group lasso problem on an infinite set of stable atoms, only a finite discretization of the atomic set is considered for tractability. This leads to an approximation error, which can only be reduced with an extensive collection of atoms (Shah et al., 2012). In addition, a significant bias is induced by lasso-type regularization (Pillonetto et al., 2016), and the pole location estimation contains a possibly large number of false positives due to the “p-value lottery" in high-dimensional regression (Meinshausen et al., 2009).

In Chapter 2, an infinite-dimensional sparse learning algorithm is addressed to tackle the above drawbacks. This algorithm directly targets the group lasso problem with an infinite feature set, which has been studied in the machine learning literature (Rakotomamonjy et al., 2012; Rosset et al., 2007; Yen et al., 2014). Two strategies based on iteratively reweighted adaptive group lasso (Wang and Leng, 2008; Gasso et al., 2009) and complementary pairs stability selection (CPSS) (Bühlmann and van de Geer, 2011; Shah and Samworth, 2013) are further presented to debias the estimate and reject false positives, respectively.

### 1.2.2 Kernel-Based Identification

Following the seminal work in Pillonetto and De Nicolao (2010), kernel-based identification (Pillonetto et al., 2014; Chiuso and Pillonetto, 2019; Ljung et al., 2019; Pillonetto et al., 2022) has received significant attention. In its basic form, a truncated impulse response model is identified with a weighted ridge regularization term. The kernel-based method can be interpreted as function learning in an RKHS, GP regression, or ridge regression with basis expansions.

The performance of this approach depends heavily on the choice of kernels, which need to be carefully designed. A class of kernel structures, such as stable spline (SS) kernels, has been proposed for system identification, which leverages, among others, the prior knowledge of stability and low complexity in system theory (Chen, 2018). This kernel design step poses problems similar to model structure selection in the classical paradigm, where the parameters in the kernel structures are the hyperparameters. Several approaches have been proposed in the literature to estimate the hyperparameters, such as the empirical Bayes method (Pillonetto

et al., 2014) and generalized cross-validation (Mu et al., 2018a,b). This process is known as hyperparameter tuning. When properly tuned, kernel-based identification can obtain more accurate nominal estimates compared to classical approaches (Pillonetto et al., 2022).

Kernel-based identification controls model complexity through the norm of the impulse response induced by an arbitrary RKHS (Chen et al., 2012). Such complexity measures need clear interpretations in classical system theory in general. In the first part of Chapter 3, a new kernel structure is introduced, which controls the number of poles in the model similar to the atomic norm regularization. This kernel structure uses multiple regularization, parametrizing the kernel in terms of basis regularization matrices with a simple design. By choosing the optimal regularization matrix for first-order systems as bases, the number of poles is bounded by the cardinality of the hyperparameters. This imposes the low-complexity feature on the identified model while maintaining the advantage of Bayesian regularization in terms of a favorable bias-variance trade-off, compared to, for example, $l_1$-norm regularization (Chen et al., 2014; Pillonetto et al., 2016).

The GP interpretation provides the kernel-based method with another advantage: it obtains Gaussian stochastic models and thus high-probability error bounds simultaneously with the nominal model (Chen et al., 2012). This enables its application in robust and stochastic control.

However, one often neglected aspect of kernel-based identification is that the results, including the error bounds, are conditioned on correct hyperparameter selection, in the same way as PEM is conditioned on the correct model structure. The hyperparameters are usually selected separately and used in identification empirically with certainty equivalence. This makes the GP-based error bounds unreliable when the estimated hyperparameters are inaccurate and thus detrimental to use in safety-critical applications. This phenomenon has been observed in machine learning literature (Rasmussen and Williams, 2006).

The second part of Chapter 3 demonstrates that the error bounds derived from estimated hyperparameters can be inaccurate in linear system identification, especially for lightly damped systems and in low signal-to-noise ratio (SNR) scenarios. Then, probabilistic error bounds are provided for kernel-based linear system identification without prior knowledge of the hyperparameters. This is done by deriving a high-probability set for the true hyperparameters and constructing error bounds for the worst case within the set.

## 1.3 From System Identification to Data-Driven Control

The paradigm of system identification has been successful in numerous control applications, enabling the design of simple but effective feedback control laws. However, the system identification step can take much work in practice. In particular, a low-dimensional model structure suitable for designing compact, closed-form control strategies can be complicated to obtain for complex systems. It constitutes the majority of the budget in model-based control design, in terms

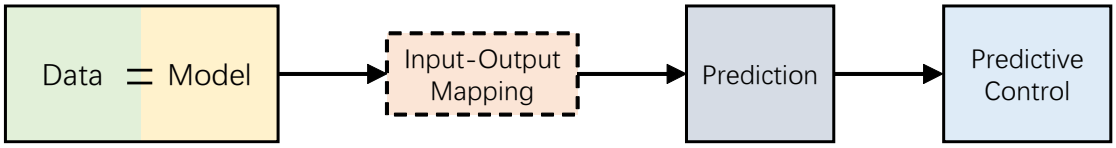



Figure 1.2: Paradigm of data-driven control.

of both time and cost (Hjalmarsson, 2005). This problem has become more prominent in recent years as the scale and complexity of the systems are increasing drastically. In addition, it becomes less clear if low-dimensional descriptions are always helpful for large-scale intelligent systems. Evidence has shown that low dimensionality is not required in modern optimization-based control frameworks and may limit the predictive power of big data (Sutton, 2019). Control design for such systems relies more on collecting abundant data rather than selecting suitable model structures and/or prior model knowledge. On the other hand, following its remarkable success in artificial intelligence, learning from data using pure black-box approaches in machine learning, such as neural networks, is becoming a popular topic in various engineering domains (Hou and Wang, 2013).

Therefore, with the availability of a massive amount of data and the advancement in computational capability, the idea of *data-driven control* has drawn significant attention. Early attempts in this direction include unfalsified control (Safonov and Tung-Ching Tsao, 1997), iterative feedback tuning (Hjalmarsson et al., 1998), and virtual reference feedback tuning (Campi et al., 2002). Reinforcement learning techniques are also widely applied in this area, including policy search (Lagoudakis and Parr, 2003) and approximate dynamic programming (Powell, 2007). However, it has been observed in Recht (2019) that these approaches tend to perform much less data-efficiently in simple tasks than model-based methods. Such approaches typically avoid predicting the behavior of systems explicitly but aim at the control strategy directly.

In this thesis, the prediction of system behaviors is still desired, but a data-driven predictor replaces the conventional parametric model. Although the paradigm of system identification can also be seen as providing indirect data-driven prediction through models, modern data-driven prediction approaches propose to skip the step of finding a low-dimensional parametrization of the system behaviors and use data directly to predict future system responses via a potentially implicit input-output mapping (van Waarde et al., 2020; Markovsky and Dörfler, 2021). In other words, the whole dataset now serves as an over-parametrized model to provide predictions for control design. This scheme is illustrated in Figure 1.2. This strategy lies in between the model-based and the so-called model-free approaches.

In this regard, a seminal result, known as the Willems' fundamental lemma (WFL) (Willems et al., 2005), shows that data-driven prediction can be conducted by linearly combining historical trajectories with sufficiently informative data for linear systems. A more general version of the lemma was recently given in Markovsky and Dörfler (2023). The matrix that collects the historical trajectories is dubbed the signal matrix. With this result, possible trajectories of the

system can be characterized by selecting a suitable combination of collected trajectories that satisfies the initial condition constraints (Markovsky and Rapisarda, 2008; De Persis and Tesi, 2020; van Waarde et al., 2020). Such characterization thus acts as a surrogate for conventional models to describe possible system trajectories. This implicit representation of the system has been successfully applied to simulation and control problems with noise-free data in Markovsky and Rapisarda (2008); Coulson et al. (2019); De Persis and Tesi (2020).

In Part II of the thesis, the extension of such data-driven trajectory predictors to stochastic data is studied in Chapter 4, whereas its application to receding horizon predictive control is discussed in Chapter 5.

### 1.3.1 Stochastic Data-Driven Trajectory Prediction

It is well-known that when data are noisy, over-parametrized models may lead to high variances and overfitting (Geman et al., 1992). In this case, finding a linear combination of collected trajectories that give reliable prediction is an ill-defined problem for datasets with stochastic noise. This issue has become the central question in data-driven approaches based on the WFL (van Waarde et al., 2022).

Two directions have been pursued in the literature to tackle the data-driven prediction problem with stochastic data. The first direction proposes to recover the low-rank structure in the noise-free data matrix which guarantees the well-definedness of the predictor. In other words, the stochastic trajectory prediction problem is converted to a low-rank matrix denoising problem. This approach has a close connection with the subspace identification approach. In fact, with a low-rank approximation, it directly leads to the intersection algorithm in subspace identification where state-space models can be derived (Moonen et al., 1989).

In the first part of Chapter 4, the low-rank matrix denoising problem with a generalized Hankel structure is investigated, where the underlying low-rank matrix is assumed to be a transformed Hankel matrix. This structure is commonly used in constructing the signal matrix. By enforcing structural constraints and avoiding approximating the noise matrix, a novel matrix denoising algorithm, which performs better than existing algorithms in terms of noise reduction, is proposed.

The second direction uses regularizers to penalize unreliable predictions, either at the predictor or control design levels. In particular, empirical regularizers (Berberich et al., 2021; Coulson et al., 2019; Dörfler et al., 2023; Lian et al., 2023) or least-norm problems, known as the data-driven subspace predictor (Favoreel et al., 1999; Huang et al., 2019; Sedghizadeh and Beheshti, 2018), have been introduced to provide reasonable predictions. Yet, the optimal design of the predictor needs to be clarified. In addition, the hyperparameters in the empirical regularizers are challenging to tune (Huang et al., 2019). The performance is often assessed with an oracle of the optimal regularization parameters by trial and error, which is unrealistic in practical applications.

In the second part of Chapter 4, an extension of the WFL to stochastic data is presented using

MLE from a system identification point of view. The derived data-driven input-output mapping dubbed the signal matrix model (SMM), provides a statistical approach to construct the stochastic data-driven predictor without sophisticated tuning. The SMM predictor can be directly used to simulate the system response (Markovsky et al., 2005b; Carapia et al., 2020). The main advantage of applying this approach is that it gives correct estimates in the noise-free case without transient or truncation errors. This is a much-desired property yet fails to be satisfied by many classical system identification methods, including least-squares regression (Chen et al., 2012) and empirical transfer function estimation (ETFE) (Ljung, 1999).

In addition to the nominal prediction, a reliable uncertainty model of the prediction is essential to robust or stochastic control design for safety-critical applications. This is, however, challenging to obtain due to the over-parametrized and implicit predictor structure with uncertainties on both the historical trajectories and the prediction conditions. In the third part of Chapter 4, a statistical framework is established to assess the prediction accuracy by providing confidence regions for a general form of stochastic data-driven predictors.

### 1.3.2 Data-Driven Predictive Control

The data-driven predictor based on the WFL is especially suitable for optimal trajectory tracking. In this regard, model predictive control (MPC) is very effective when an accurate system model is available (Kouvaritakis and Cannon, 2016). The output predictor using an explicit parametric model in MPC can be replaced by the nonparametric data-driven predictor discussed above. This data-driven alternative to MPC algorithms, known as data-driven predictive control (DDPC), has led to multiple successful algorithms, including subspace predictive control (SPC) (Favoreel et al., 1999; Kadali et al., 2003; Hallouzi and Verhaegen, 2008; Sedghizadeh and Beheshti, 2018), data-enabled predictive control (DeePC) (Huang et al., 2019; Coulson et al., 2019; Coulson et al., 2022; Alpago et al., 2020), and behavioral input-output parametrization (Furieri et al., 2021) with stability and robustness proofs given in Berberich et al. (2021). Successful applications have been found in different fields, including power systems (Huang et al., 2021; Huang et al., 2019), quadrotors (Elokda et al., 2021), and building control (Lian et al., 2023).

Chapter 5 focuses on the application of the SMM to DDPC algorithms. The main advantage of using SMM is that it avoids the difficult hyperparameter tuning problem in existing DDPC algorithms while obtaining a better performance numerically. With the prediction error quantification in Chapter 4, a stochastic version of the SMM-based DDPC algorithm is proposed with the following novelties: 1) initial condition estimation by Kalman filtering, 2) chance constraint satisfaction by constraint tightening, and 3) regularization based on the stochastic cost. Finally, high-fidelity simulation results are presented in a space heating control example.

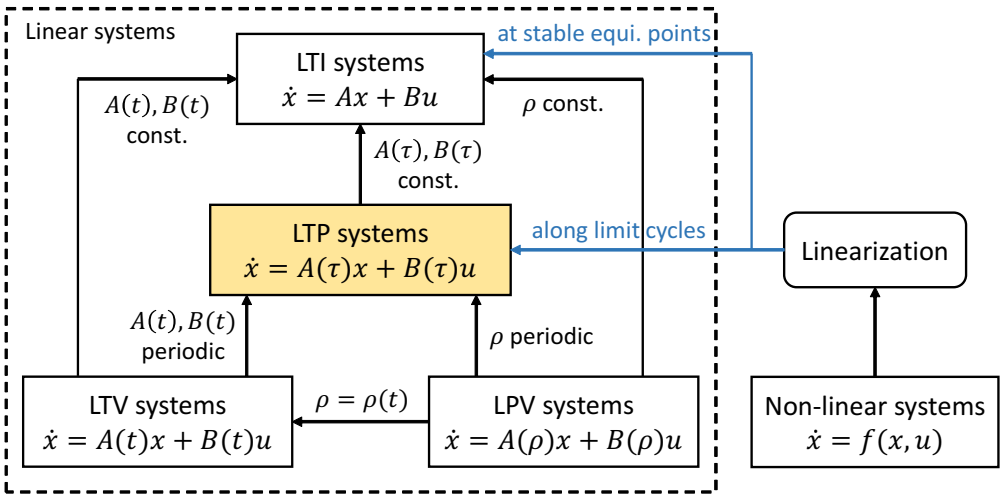



Figure 1.3: Relations between periodic systems and other types of systems.

## 1.4 Periodic Systems

In addition to linear time-invariant (LTI) systems considered in the first two parts of the thesis, we are also interested in identifying periodic systems from input-output data. Periodicity in dynamics, scheduling parameters, and operating trajectories is often observed in various applications, including rotating machinery (Allen et al., 2011), aerospace (Shin et al., 2005; Wood et al., 2018), power systems (Möllerstedt and Bernhardsson, 2000), building control (Khosravi et al., 2017), and process control (Budman and Silveston, 2013). More importantly, periodic models serve as an intermediate step to capture more general representations than LTI systems, such as linear time-varying (LTV) systems (Liu, 1997), linear parameter-varying (LPV) systems (Felici et al., 2007; Goos and Pintelon, 2014; Cox, 2018) and nonlinear systems along limit cycles (Allen and Sracic, 2009). The relations between periodic systems and other types of systems are illustrated in Figure 1.3.

In Chapter 6, two approaches are proposed to identify linear time-periodic (LTP) systems using the regularized method in the time domain and the subspace method in the frequency domain, respectively. LTP systems are systems with periodically time-varying linear dynamics. In general, any identification scheme for LTI systems applies to LTP systems by application of the lifting technique (Bittanti and Colaneri, 2000). However, such methods often fail to encode characteristics of the lifted system, such as the causality constraint that prevents future inputs in a period from affecting previous outputs. The identified lifted system is thus not guaranteed to be realizable to its LTP form. As pointed out in Bittanti and Colaneri (2000), the critical issue in LTP system identification is that the parameters in the reformulated LTI models have strong correlations since they come from the same dynamic system. This correlation is not investigated in existing LTI-based identification frameworks.

As discussed in Section 1.2, regularized methods have recently reported positive results in linear

system identification. For LTI reformulations of LTP systems in particular, this framework enables us to treat the structural constraints using parameter regularization. The first part of Chapter 6 extends the atomic norm regularization approach to LTP systems by using a group lasso regularizer to impose the additional structural constraints needed for periodic models. This approach estimates uniform low-order models for LTP systems.

The second part of Chapter 6 focuses on identifying state-space models of LTP systems. Most existing methods for this problem are time-domain subspace identification methods (Verhaegen and Yu, 1995), which extend naturally from its LTI counterpart (Overschee and Moor, 1996). This method, along with a similar version in Hench (1995), has contributed to several successful applications (e.g., Felici et al. (2007); Sefidmazgi et al. (2016)), especially in identifying LPV systems where modern subspace techniques have been incorporated (Cox, 2018). On the other hand, the frequency-domain subspace formulation for LTP systems has not been well-investigated.

Frequency domain methods in system identification are particularly suitable when periodic inputs are used in identification experiments. As discussed in Schoukens et al. (1994), periodic input design has several advantages compared to random input design, including avoiding initial state estimation and easier time-domain averaging. However, the frequency response behavior of LTP systems differs significantly from that of LTI systems (Wereley, 1990). Most prominently, the independence of the frequency response at different frequencies, a property fundamental to frequency-domain identification of LTI systems, does not hold for LTP systems. This prevents straightforwardly applying LTI techniques to frequency-domain identification of LTP systems. Instead, in this thesis, a novel frequency-domain subspace identification method for multiple-input multiple-output (MIMO) LTP systems is proposed by first estimating the frequency response of the time-lifted system with LTI structure. Then, the periodic impulse response of the original LTP system is recovered where an extension to the LTI frequency domain subspace method can be applied.

In Chapter 7, linear periodic models are adopted to identify nonlinear limit cycle dynamics. Limit cycles for nonlinear dynamical systems are periodic equilibrium orbits that, if locally stable, are local attractors and thus lead to self-sustained periodic oscillations (Strogatz, 1994). When a system is controlled along a periodic reference, the closed-loop dynamics can also be considered a limit cycle. In this regard, it is of interest to identify a model that describes the limit cycle dynamics for simulation, analysis, and iterative control design of the closed-loop system.

Identifying nonlinear systems purely from data poses a complex problem, requiring prior knowledge of the model structure and/or complex nonlinear optimization schemes with tractability issues (Schoukens and Ljung, 2019). Instead, local linear dynamics are often identified for different operating points to construct an LPV model with gain scheduling applied in control design (Tóth, 2010). The local dynamics close to the limit cycles are often of primary concern for limit cycles. However, conventional LPV methods do not consider that the underlying model converges to a limit cycle.

In this thesis, an alternative approach that identifies the nonlinear dynamics around the limit cycle as a linear periodically parameter-varying (LPPV) model is investigated. This approach decomposes the dynamics into two parts: one moving along the limit cycle and one lying on the transversal hyperplanes of the limit cycle, known as Poincaré sections. These two parts can both be modeled as locally linearized LPPV models. The linearized transverse dynamics reduce the nonlinear identification problem to learning the periodic system matrices as functions of the location on the limit cycle. This function learning problem is tackled using kernel methods in an LPV system identification framework (Bachnas et al., 2014) with periodic kernel design. The algorithm is applied to a simplified kinematic model of a tethered kite for airborne wind energy generation (Ahrens et al., 2013).

## 1.5 Main Problem Formulation

Unless otherwise specified, in the first two parts of the thesis, we consider a causal and stable LTI single-input single-output (SISO) discrete-time system. It can be written in the transfer function form

$$y_t = G_0(q)u_t + v_t, \tag{1.1}$$

the state-space form

$$\begin{cases} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t + Du_t + v_t, \end{cases} \tag{1.2}$$

or the infinite impulse response (IIR) form

$$y_t = \sum_{l=0}^{\infty} g_l u_{t-l} + v_t, \quad G_0(q) = \sum_{l=0}^{\infty} g_l q^{-l}, \tag{1.3}$$

where $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$, $y_t \in \mathbb{R}^{n_y}$, $v_t \in \mathbb{R}^{n_y}$ are the states, inputs, outputs, and output noise respectively, $q$ is the forward time-shift operator, and $g_l$ is the impulse response of the system. The additive noise is assumed to be zero-mean independent and identically distributed (i.i.d.) Gaussian with a variance of $\sigma^2$, i.e., $v_t \sim \mathcal{N}(0, \sigma^2\mathbb{I})$. The system is assumed to be observable with observability index (lag) $l$. The noise-free output is denoted by $y_{t,0}$.

An input-output sequence of the system

$$\mathbf{u}^d := \begin{bmatrix} u_1^d & u_2^d & \dots & u_N^d \end{bmatrix}^\top, \quad \mathbf{y}^d := \begin{bmatrix} y_1^d & y_2^d & \dots & y_N^d \end{bmatrix}^\top \tag{1.4}$$

has been collected, where the superscript $d$ denotes data.

In Part I, we are interested in identifying the model $G_0(q)$ from the data sequence $(\mathbf{u}^d, \mathbf{y}^d)$ using regularized system identification techniques.

In regularized system identification, the transfer function $G_0(q)$ is expressed with a general high-dimensional parametrization $G_0(q) = \sum_{k\in\mathcal{K}} c_k A_k(q)$, where $A_k(q)$ are the basis transfer

functions, $c_k$ are the corresponding coefficients, and $\mathcal{K}$ denotes the set of indices. Denote the set of coefficients as $C := \{c_k | k \in \mathcal{K}\}$.

In Chapter 2, we consider stable first-order atoms (Shah et al., 2012)

$$A_k(q) := \frac{1-|k|^2}{q-k}, \quad c_k \in \mathbb{C} \tag{1.5}$$

and a static atom $A_1(q) := 1$ to accommodate non-zero feedthrough terms. The coefficient set is given by $\mathcal{K} = \{k = \alpha \exp(j\beta) \,|\, \alpha \in [0,1), \beta \in [0, 2\pi)\} \cup \{1\}$. This selection of atoms also guarantees the stability of the estimated system. The atoms $A_k(q)$ are normalized to have a Hankel nuclear norm of 1.

In Chapter 3, we consider the impulse response model, i.e.,

$$A_k(q) := q^{-k}, \quad c_k \in \mathbb{R}, \quad \mathcal{K} = \mathbb{Z}_+. \tag{1.6}$$

The following regularized optimization problem is solved:

$$\min_{C} \quad \mathcal{L}\left(\mathbf{y}^d - \sum_{k\in K} c_k \phi\left(A_k(q), \mathbf{u}^d\right)\right) + \lambda \mathcal{R}(C), \tag{1.7}$$

where $\phi\left(A(q), \mathbf{u}^d\right)$ denotes the length-$N$ output response of the system $A(q)$ to the inputs $\mathbf{u}^d$, $\mathcal{L}(\cdot)$ is the loss function that penalizes the output residuals, $\mathcal{R}(\cdot)$ is the regularizer that encodes prior knowledge of the coefficients, and $\lambda$ is the regularization parameter to tune the amount of regularization. In this thesis, the loss function is selected as $\mathcal{L}(x) := \|x\|_2^2$, which is related to MLE when the noise $v_t$ is zero-mean i.i.d. Gaussian.

The accuracy of the estimated model can be assessed by evaluating the fitting of the estimated impulse response to the true impulse response within a length of $n_g$, defined as

$$W := 100 \cdot \left(1 - \left[\frac{\sum_{l=0}^{n_g-1}(g_l - \hat{g}_l)^2}{\sum_{l=0}^{n_g-1}(g_l - \bar{g})^2}\right]^{1/2}\right), \tag{1.8}$$

where $\hat{g}_l$ are the estimated impulse responses, and $\bar{g}$ is the mean of $(g_l)_{l=0}^{n_g-1}$. This measure is equivalent to the `compare` function in MATLAB.

In Part II, instead of obtaining the model explicitly, we focus on obtaining a nonparametric input-output mapping for predicting system responses directly from data sequences $(\mathbf{u}^d, \mathbf{y}^d)$. In detail, we are interested in predicting the length-$L'$ output trajectory $\mathbf{y} := \mathrm{col}\,(y_0, \ldots, y_{L'-1})$ from any given input trajectory $\mathbf{u} := \mathrm{col}\,(u_0, \ldots, u_{L'-1})$ using only $(\mathbf{u}^d, \mathbf{y}^d)$. To obtain a unique output trajectory, the initial condition is also fixed by measuring the length-$L_0$ immediate past input-output trajectory $\mathbf{u}_{\text{ini}} := \mathrm{col}\,(u_{-L_0}, \ldots, u_{-1})$ and $\mathbf{y}_{\text{ini}} := \mathrm{col}\,(y_{-L_0}, \ldots, y_{-1})$, where $L_0 \geq l$. This guarantees the uniqueness of the initial condition due to the definition of the observability

index. In other words, a nonparametric data-driven predictor is desired by obtaining the following input-output mapping:

$$\mathbf{y} = \mathcal{F}_{(\mathbf{u}^d,\mathbf{y}^d)}(\mathbf{u};\mathbf{u}_{\text{ini}},\mathbf{y}_{\text{ini}}). \tag{1.9}$$

This input-output mapping is used to design a data-driven optimal trajectory tracking controller in Chapter 5. It aims to track a reference trajectory $\mathbf{r}$ in the sense of minimizing the total control cost:

$$J_{\text{tot}} := \sum_{t=0}^{N_c-1} J_t := \sum_{t=0}^{N_c-1} \left( \|u_t\|_R^2 + \|y_{t,0} - r_t\|_Q^2 \right), \tag{1.10}$$

where $J_t$ is the stage cost at time $t$, $R, Q$ are the input and the output cost matrices, respectively, and $N_c$ is the length of the control task.

## 1.6 List of Publications

The contents of this thesis are based on the following publications.

**Chapter 2:**

- **Yin M.**, Akan M.T., Iannelli A., Smith R.S. (2022). Infinite-Dimensional Sparse Learning in Linear System Identification. *IEEE Conference on Decision and Control*.

**Chapter 3:**

- Khosravi M.*, **Yin M.***, Iannelli A., Parsi A., Smith R.S. (2020). Low-Complexity Identification by Sparse Hyperparameter Estimation. *IFAC World Congress*.
- **Yin M.**, Smith R.S. (2023). Error Bounds for Kernel-Based Linear System Identification with Unknown Hyperparameters. *IEEE Control Systems Letters*, 7, 2491-2496.

**Chapter 4:**

- **Yin M.**, Smith R.S. (2021). On Low-Rank Hankel Matrix Denoising. *IFAC Symposium on System Identification*.
- **Yin M.**, Iannelli A., Smith R.S. (2021). Maximum Likelihood Estimation in Data-Driven Modeling and Control. *IEEE Transactions on Automatic Control*, 68(1), 317-328.
- **Yin M.**, Iannelli A., Smith R.S. (2022). Data-Driven Prediction with Stochastic Data: Confidence Regions and Minimum Mean-Squared Error Estimates. *European Control Conference*.

**Chapter 5:**

- **Yin M.**, Iannelli A., Smith R.S. (2021). Maximum Likelihood Signal Matrix Model for Data-Driven Predictive Control. *Conference on Learning for Dynamics and Control*, PMLR 144:1004-1014.

- **Yin M.**, Iannelli A., Smith R.S. (2023). Stochastic Data-Driven Predictive Control: Regularization, Estimation, and Constraint Tightening. arXiv:2312.02758.
- **Yin M.**, Cai H., Gattiglio A., Khayatian F., Smith R.S, Heer P. (2024). Data-Driven Predictive Control for Demand Side Management: Theoretical and Experimental Results. *Applied Energy*, 353(Part A), 122101.

**Chapter 6:**

- **Yin M.**, Iannelli A., Khosravi M., Parsi A., Smith R.S. (2020). Linear Time-Periodic System Identification with Grouped Atomic Norm Regularization. *IFAC World Congress*.
- **Yin M.**, Iannelli A., Smith R.S. (2021). Subspace Identification of Linear Time-Periodic Systems with Periodic Inputs. *IEEE Control Systems Letters*, 5(1), 145-150.

**Chapter 7:**

- Ozan D.E., **Yin M.**, Iannelli A., Smith R.S. (2022). Kernel-Based Identification of Local Limit Cycle Dynamics with Linear Periodically Parameter-Varying Models. *IEEE Conference on Decision and Control*.

# Part I
# Regularized Methods in System Identification

# 2 Sparse Learning in System Identification

In this chapter, we first revisit the fundamental theories of sparse learning in high-dimensional statistics in Section 2.1. After introducing the basic lasso and group lasso problems, iteratively reweighted adaptive approaches (Wang and Leng, 2008; Gasso et al., 2009) are discussed to reduce the amount of regularization on significant features, thus reducing the bias. Complementary pairs stability selection (CPSS) (Bühlmann and van de Geer, 2011; Shah and Samworth, 2013) is discussed to estimate the active set stably by solving the lasso-type problem repeatedly on subsamples of the identification data and selecting features that are consistently active.

These methods are applied to the atomic norm regularization in linear system identification in Section 2.2. An infinite-dimensional group lasso problem is formulated using first-order atoms. A tractable algorithm to solve the infinite-dimensional problem is presented. It first solves the problem with a small number of randomly generated features. Then, by inspecting the optimality conditions of the finite-dimensional problem, a new atomic model feature is selected to maximize the optimality condition violation for the previous iteration. This greedy iteration repeats until no new features can be added. The algorithm guarantees a decrease in the objective value per iteration and solves the infinite-dimensional problem with an arbitrarily small tolerance. The group lasso estimate is further debiased by iterative reweighing, and reliable pole location estimation is obtained by CPSS.

Numerical results demonstrate that the proposed algorithm performs better than PEM with an ARX model, kernel-based identification with tuned/correlated (TC) kernel design, and the existing atomic norm regularization algorithm in terms of impulse response fitting on a benchmark system. In addition, adaptive group lasso can reduce the algorithm's bias, and CPSS obtains more accurate pole location estimation than PEM with fewer false positives.

## 2.1 High-Dimensional Regression

Consider linear regression problem

$$\mathbf{y} = X\mathbf{c}^0 + \mathbf{e}, \tag{2.1}$$

where $\mathbf{y} \in \mathbb{R}^N$, $X \in \mathbb{R}^{N\times n}$, $\mathbf{e} \in \mathbb{R}^N$ are the regressand, regressor, and noise respectively and $\mathbf{c}^0 \in \mathbb{R}^n$ is the true parameter to be estimated.

High-dimensional regression refers to regression problems where $n \gg N$. In general, this leads to underdetermined, ill-defined problems. To make the problem well-defined, we restrict it to sparse problems. Define

$$S(\mathbf{c}^0) := \left\{ j \,\middle|\, c_j^0 \neq 0 \right\}, \quad \#\left[S(\mathbf{c}^0)\right] := p(\mathbf{c}^0), \tag{2.2}$$

where $S(\mathbf{c}^0)$ and $p(\mathbf{c}^0)$ are known as the active set and the cardinality of $\mathbf{c}^0$, respectively. The parameter $\mathbf{c}^0$ is considered sparse if $p(\mathbf{c}^0) \ll N$. Sparse learning techniques can be applied to high-dimensional regression problems under this assumption.

### 2.1.1 Lasso and Group Lasso

Suppose the true parameter cardinality $p(\mathbf{c}^0)$ is known. It is natural to formulate the sparse learning problem as a constrained least-squares problem

$$\min_{\mathbf{c}} \ \|\mathbf{y} - X\mathbf{c}\|_2^2 \quad \text{s.t.} \quad p(\mathbf{c}) = p(\mathbf{c}^0). \tag{2.3}$$

However, the cardinality constraint is NP-hard, and $p(\mathbf{c}^0)$ is usually unknown. Thus, the cardinality function is often replaced with the $l_1$-norm, the best convex surrogate for the cardinality function. Rewrite the $l_1$-norm constrained problem to its Lagrangian form, we have

$$\min_{\mathbf{c}} \quad \|\mathbf{y} - X\mathbf{c}\|_2^2 + \lambda \mathcal{R}_{\text{lasso}}(\mathbf{c}), \quad \mathcal{R}_{\text{lasso}}(\mathbf{c}) := \|\mathbf{c}\|_1, \tag{2.4}$$

which is the well-known lasso problem.

An important extension to lasso is its grouped variant known as group lasso (Yuan and Lin, 2006), where sparsity is enforced on groups of parameters rather than isolated scalar parameters. Consider a grouping of parameter $\mathbf{c} =: \operatorname{col}\left(\mathbf{c}_1, \ldots, \mathbf{c}_{n_{\text{gl}}}\right)$: $\mathbf{c}_i \in \mathbb{R}^{n_i}, i = 1, 2, \ldots, n_{\text{gl}}$. The group lasso regularizer is given by

$$\mathcal{R}_{\text{glasso}}(\mathbf{c}) := \sum_{i=1}^{n_{\text{gl}}} \|\mathbf{c}_i\|_2 . \tag{2.5}$$

Here, $l_2$-norms are used to relax the sparsity constraint inside each group, and the sparsity-promoting function reduces to summation since the $l_2$-norms are always non-negative. In this way, sparsity is enforced on the group $l_2$-norms: when the $l_2$ norm is regularized to zero, all parameters in the group are zero; when the $l_2$ norm is non-zero, all the parameters are usually non-zero. So, consistent sparsity is promoted inside each group.

Lasso and group lasso methods have been widely used in system identification, including identification of switched systems (Ohlsson and Ljung, 2013), dynamic networks (Chiuso and Pillonetto, 2012), and non-linear systems with heterogeneous data (Pan et al., 2018). In this thesis, we focus on their application to atomic norm regularization.

### 2.1.2 Debiasing by Iteratively Reweighted Adaptive Lasso

The $l_1$-norm regularizer is a convex relaxation of the ideal sparsity promoting function $\mathcal{R}_{\text{ideal}}(\mathbf{c}) := \#[S(\mathbf{c})]$. Compared to the ideal regularizer, which penalizes all the non-zero parameters with a fixed value of 1, the $l_1$-norm regularizer penalizes them with the magnitude of the corresponding coefficients. This induces a bias, especially for larger parameter values, i.e., the dominant modes. This bias is a significant source of error in lasso (Pillonetto et al., 2016).

To reduce such bias, adaptive lasso (Zou, 2006) has been proposed, which adds a second step that applies a reweighted version of the $l_1$-norm regularizer

$$\mathcal{R}_{\text{alasso}}(\mathbf{c}) := \sum_{i=1}^{n} \frac{|c_i|}{|c_i^\star| + \varepsilon}, \tag{2.6}$$

where $c_i^\star$ is the solution to the original lasso problem, and $\varepsilon > 0$ is a small constant to avoid singularity. This regularizer reduces the amount of regularization for large coefficients in the original problem and is close to $\mathcal{R}_{\text{ideal}}(\mathbf{c})$ when $c_i \approx c_i^\star$.

This approach is extended to apply this reweighting iteratively (Bühlmann and van de Geer, 2011, Section 2.8.5) by using the solution at the last iteration to update $c_i^\star$. This is sometimes known as iteratively reweighted lasso. It is pointed out in Gasso et al. (2009) that the iteratively reweighted lasso can be interpreted as a difference of convex programming algorithm to solve the regularized problem with a non-convex logarithmic regularizer

$$\mathcal{R}_{\log}(\mathbf{c}) := \sum_{i=1}^{n} \frac{\log\left(|c_i| + \varepsilon\right)}{\log \varepsilon}. \tag{2.7}$$

**Example 2.1.** *(Bias in lasso-type algorithms) Consider an identity regressor $X = \mathbb{I}$, $N = n$. The solutions of the estimate $\hat{\mathbf{c}}$ can be derived analytically for regularizers* (2.4)*,* (2.6)*, and* (2.7) *as functions of $\hat{c}_i$ with respect to $y_i$. These functions are plotted in Figure 2.1 with $\lambda = 2$. As shown in Figure 2.1, the lasso estimates the parameter with a soft-thresholding function, leading to a constant bias for high $y_i$-values. This bias is mitigated with the adaptive and the logarithmic modifications.*

The iteratively reweighted adaptive lasso can also be naturally extended to group lasso (Wang and Leng, 2008), where the regularizer becomes

$$\mathcal{R}_{\text{galasso}}(\mathbf{c}) := \sum_{i=1}^{n_{\text{gl}}} \frac{\|\mathbf{c}_i\|_2}{\|\mathbf{c}_i^\star\|_2 + \varepsilon}. \tag{2.8}$$

### 2.1.3 Variable Selection by Stability Selection

Lasso-type regularized problems are known to have favorable consistency properties in terms of prediction under mild conditions. However, in terms of estimating the true active set $S(\mathbf{c}^0)$,

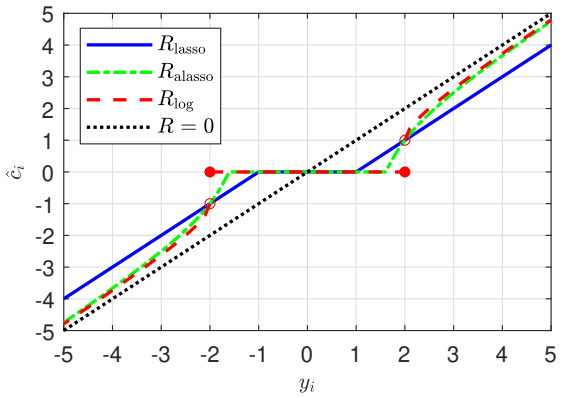



Figure 2.1: Illustration of the shrinkage property of the lasso, adaptive lasso, and logarithmic regularizers.

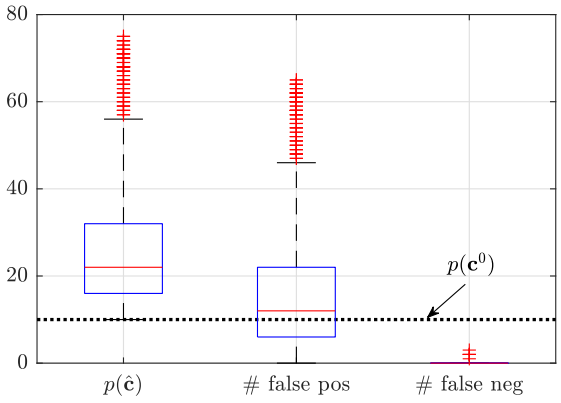



Figure 2.2: Illustration of the variable selection property of the lasso estimator.

they can only guarantee that the non-active parameters are not in the true model with high probability under practical assumptions (Bühlmann and van de Geer, 2011, Chapter 2). This property is known as variable screening. Only the number of false negatives in the estimated active set is controlled, but not that of false positives. In fact, there are usually many more non-zero parameters in the estimate than those in the true one, with many occurring at "random" locations depending on the noise realization. This phenomenon is known as the "p-value lottery" (Meinshausen et al., 2009).

**Example 2.2.** *(False positives in variable selection) A total of 1000 Monte Carlo simulations are conducted with* $N = 80$*,* $n = 1000$*,* $\mathbf{c}^0 = \operatorname{col}(\mathbf{1}_{10}, \mathbf{0}_{990})$*,* $p(\mathbf{c}^0) = 10$*,* $e_i \sim \mathcal{N}(0, 0.1)$*. A random design of* $X$*:* $X_{i,j} \sim \mathcal{N}(0,1)$ *is used. The regularization parameters* $\lambda$ *are selected by cross-validation. Figure 2.2 plots the boxplot of* $p(\hat{\mathbf{c}})$ *with lasso as well as the number of false positives and negatives. It becomes clear that while lasso controls the false negatives well, it overestimates the active set with many false positives.*

If we could repeat the experiment many times, it would become obvious that the true active parameters occur repeatedly in the estimated active set, whereas the false positives only occur

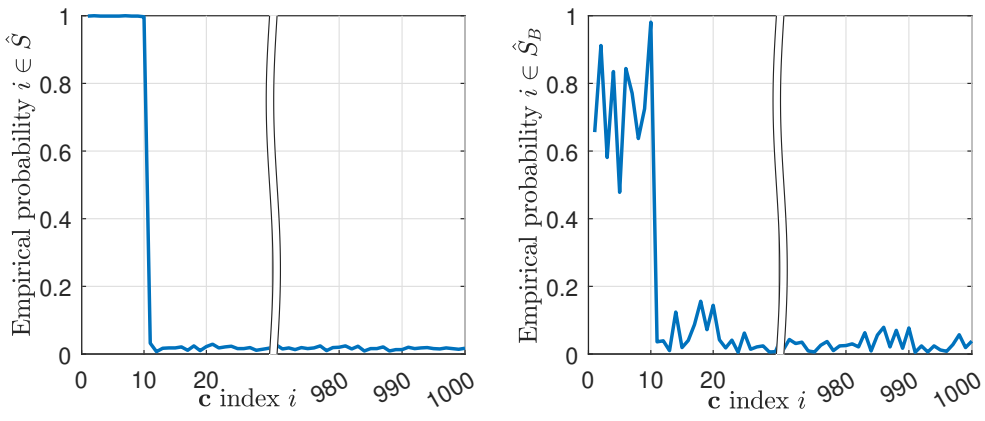



Figure 2.3: Stability selection with (a) independent experiments and (b) subsamples.

randomly. Inspired by this idea, subsampling techniques have been used to generate new artificial "experiments" to increase the stability of the active set estimation. One such algorithm is CPSS proposed in Shah and Samworth (2013). This method generates complementary pairs of subsamples from the identification data and repeats the baseline variable selection procedure (lasso-type problems in this case) on each subsample. Define the estimated active set on the subsample as $\hat{S}_B$, where $B \subset \{1,2,\ldots,N\}$ defines a random subsample of data. Then, the so-called stable solution to the problem is defined as the parameters with higher empirical probabilities of being included in $\hat{S}_B$ than a predefined threshold $\tau$. The algorithm has favorable false-positive rejection properties when $\tau > 0.5$ (Shah and Samworth, 2013).

**Example 2.3.** *(Stability selection with subsamples) Consider the same problem as in Example 2.2. Figure 2.3(a) plots the empirical probabilities of each parameter being active in the 1000 Monte Carlo simulations, whereas Figure 2.3(b) plots the empirical probabilities in 1000 subsamples from only one simulation. Although not as clear as using 1000 independent simulations, the active parameters can still be selected with high empirical probabilities by subsampling.*

## 2.2 Atomic Norm Regularization for Model Complexity Control

One of the critical aspects of moving towards high-dimensional models is to control the model complexity of the identified model in accordance with the principle of parsimony and to avoid overfitting. In classical system identification, this is done by controlling the number of poles of the system, which is still a common objective here despite the use of general high-dimensional models. The Hankel nuclear norm of the impulse response is used as a convex surrogate in Smith (2014); Fazel et al. (2001). However, this regularizer is prone to stability (Pillonetto et al., 2016) and scalability (Shah et al., 2012) issues.

Instead, we note that, for atoms (1.5), the number of poles equals the number of non-zero elements in $C$. In addition, the pole locations of the system are directly given by the active set $S(C) := \{k \,|\, |c_k| > 0\}$, with slight abuse of notation. Unlike what is discussed in Section 2.1, here

$k$ is a stable pole within the open unit disk, so the set of indices $\mathcal{K}$ thus has infinite elements. The coefficients $c_k$ are also complex.

**Remark 2.1.** *The atomic decomposition with* (1.5) *does not exactly cover the case where you have repeated poles. However, with a continuous set of atoms, repeated poles can be approximated by adjacent poles with arbitrarily high accuracy.*

To control the number of poles of the system, a sparsity-promoting regularization term $\mathcal{R}(C)$ is desired. As discussed in Section 2.1, a tractable $l_1$-norm regularizer

$$\mathcal{R}(C) = \sum_{k \in K} |c_k| \tag{2.9}$$

is used and defined as the atomic norm of the model with respect to the atoms (1.5) (Tibshirani, 1996).

### 2.2.1 Real-Valued Reformulation

With atoms (1.5), the coefficients $c_k$ are complex-valued, which leads to complex-valued program (1.7). Observe that for real-rational systems, the pole locations should be in conjugate pairs and the corresponding atomic responses are also complex conjugates of one another, i.e., $\phi\left(A_{\bar{k}}(q), \mathbf{u}^d\right) = \bar{\phi}\left(A_k(q), \mathbf{u}^d\right)$, where the overbar denotes the complex conjugate. This means that coefficients for a conjugate pole pair should also be complex conjugates, i.e., $c_{\bar{k}} = \bar{c}_k$. Adding this constraint on the coefficients of (1.7), the problem can be reformulated as

$$\min_{\{c_k\}_{k \in \hat{\mathcal{K}}}} \quad \left\| \mathbf{y}^d - \sum_{k \in \hat{\mathcal{K}}} \left( c_k \phi_k + \bar{c}_k \bar{\phi}_k \right) \right\|_2^2 + 2\lambda \sum_{k \in \hat{\mathcal{K}}} |c_k|, \tag{2.10}$$

where $\phi_k := \phi\left(A_k(q), \mathbf{u}^d\right)$ for notational convenience and

$$\hat{\mathcal{K}} := \{k = \alpha \exp(j\beta) \,|\, \alpha \in [0,1), \beta \in [0,\pi]\} \cup \{1\} \tag{2.11}$$

denotes the upper half of the open unit disk and the static atom.

Let

$$\gamma_k := \begin{bmatrix} \Re(c_k) & \Im(c_k) \end{bmatrix}^\top, \; \zeta_k := \begin{bmatrix} 2\Re(\phi_k) & -2\Im(\phi_k) \end{bmatrix}. \tag{2.12}$$

Substituting (2.12) into (2.10), (2.10) can be expressed as a real-valued problem,

$$\Gamma^\star := \{\gamma_k^\star\}_{k \in \hat{\mathcal{K}}} = \underset{\{\gamma_k\}_{k \in \hat{\mathcal{K}}}}{\operatorname{argmin}} \underbrace{\left\| \mathbf{y}^d - \sum_{k \in \hat{\mathcal{K}}} \zeta_k \gamma_k \right\|_2^2 + 2\lambda \sum_{k \in \hat{\mathcal{K}}} \|\gamma_k\|_2}_{J(\Gamma)}, \tag{2.13}$$

where $\Gamma := \left\{ \gamma_k \,|\, k \in \hat{\mathcal{K}} \right\}$. Note that (2.13) is a standard group lasso problem (Yuan and Lin,

2006). The identified transfer function can be recovered by

$$\hat{G}(q) = \sum_{k \in \hat{\mathcal{K}}} [1 \;\; j]\, \gamma_k^\star A_k(q) + [1 \;\; -j]\, \gamma_k^\star A_{\bar{k}}(q), \tag{2.14}$$

and the estimated pole locations are

$$\hat{S} = \{k \,|\, \|\gamma_k^\star\|_2 > 0\} \cup \{\bar{k} \,|\, \|\gamma_k^\star\|_2 > 0\}. \tag{2.15}$$

However, problem (2.13) cannot be directly solved since it is an infinite-dimensional problem. Existing algorithms relax this problem by approximating $\hat{\mathcal{K}}$ with a discrete grid (Shah et al., 2012). As shown in Proposition 4.1 of Shah et al. (2012), the discretization induces a relative error in the atomic norm that is inversely proportional to the square root of the number of elements in the discretized $\hat{\mathcal{K}}$.

### 2.2.2 Algorithm for Infinite-Dimensional Atomic Norm Regularization

This subsection proposes an algorithm to solve the infinite-dimensional problem (2.13) directly. This algorithm is inspired by the feature generation algorithm in Rakotomamonjy et al. (2012).

Problem (2.13) is a non-differentiable convex program whose optimality conditions are given by $0 \in \partial J(\Gamma)$. In detail, the optimality conditions of (2.13) are

$$\begin{cases} \left\|\zeta_k^\top R\right\|_2 \le \lambda, & \text{if } \left\|\gamma_k^\star\right\|_2 = 0, \\ \zeta_k^\top R + \lambda\, \gamma_k^\star / \left\|\gamma_k^\star\right\|_2 = 0, & \text{if } \left\|\gamma_k^\star\right\|_2 > 0, \end{cases} \tag{2.16}$$

for all $k \in \hat{\mathcal{K}}$, where $R := \mathbf{y}^d - \sum_{k \in \hat{\mathcal{K}}} \zeta_k \gamma_k^\star$ is the vector of output residuals. The derivation makes use of the property

$$\partial \|\gamma_k^\star\|_2 = \begin{cases} \{w \,|\, \|w\|_2 \le 1\}, & \left\|\gamma_k^\star\right\|_2 = 0, \\ \gamma_k^\star / \left\|\gamma_k^\star\right\|_2, & \left\|\gamma_k^\star\right\|_2 > 0. \end{cases} \tag{2.17}$$

Let $\hat{\mathcal{K}}_d := \{k_1, k_2, \ldots, k_p\}$ be a finite subset of $\hat{\mathcal{K}}$ with $p$ elements. Then, with slight abuse of notation, by replacing $\hat{\mathcal{K}}$ with $\hat{\mathcal{K}}_d$ in (2.13), a discretized optimal solution, denoted by $\Gamma^\star(\hat{\mathcal{K}}_d) := \left\{\gamma_i^\star(\hat{\mathcal{K}}_d)\right\}_{i=1}^p$, can be obtained, which satisfies

$$\begin{cases} \left\|\zeta_i(\hat{\mathcal{K}}_d)^\top R(\hat{\mathcal{K}}_d)\right\|_2 \le \lambda, & \text{if } \left\|\gamma_i^\star(\hat{\mathcal{K}}_d)\right\|_2 = 0, \\ \zeta_i(\hat{\mathcal{K}}_d)^\top R(\hat{\mathcal{K}}_d) + \lambda \dfrac{\gamma_i^\star(\hat{\mathcal{K}}_d)}{\left\|\gamma_i^\star(\hat{\mathcal{K}}_d)\right\|_2} = 0, & \text{if } \left\|\gamma_i^\star(\hat{\mathcal{K}}_d)\right\|_2 > 0, \end{cases} \tag{2.18}$$

for $i = 1, \ldots, p$, where $R(\hat{\mathcal{K}}_d) := \mathbf{y}^d - \sum_{i=1}^p \zeta_i(\hat{\mathcal{K}}_d)\gamma_i^\star(\hat{\mathcal{K}}_d)$ and $\zeta_i(\hat{\mathcal{K}}_d) := \zeta_{k_i}$.

Suppose we want to add a new element $k_{p+1}$ to $\hat{\mathcal{K}}_d$. Then the optimal solution with respect to

$\hat{\mathcal{K}}_d^+ := \hat{\mathcal{K}}_d \cup \{k_{p+1}\}$ is

$$\gamma_i^\star(\hat{\mathcal{K}}_d^+) = \begin{cases} \gamma_i^\star(\hat{\mathcal{K}}_d), & i = 1,\dots,p, \\ \mathbf{0}, & i = p+1, \end{cases} \tag{2.19}$$

iff $\left\| \zeta_{p+1}(\hat{\mathcal{K}}_d^+)^\top R(\hat{\mathcal{K}}_d) \right\|_2 \le \lambda$. In other words, adding such new elements does not improve the optimal objective function value or change the transfer function estimate $\hat{G}(q)$. So the new element only reduces the objective function value when $\left\| \zeta_{k_{p+1}}^\top R(\hat{\mathcal{K}}_d) \right\|_2 > \lambda$. This also guarantees $k_{p+1} \notin \hat{\mathcal{K}}_d$ since $\left\| \zeta_{k_i}^\top R(\hat{\mathcal{K}}_d) \right\|_2 \le \lambda$ for $i = 1,\dots,p$.

Motivated by the above observation, Algorithm 2.1 is proposed to solve the infinite-dimensional group lasso problem (2.13), where a greedy strategy is applied that chooses the new element by maximizing $\left\| \zeta_{k_{p+1}}^\top R(\hat{\mathcal{K}}_d) \right\|_2$. Note that $\hat{\mathcal{K}}_d^l$ denotes the set $\hat{\mathcal{K}}_d$ at the $l$-th iteration in Algorithm 2.1. The transfer function and the pole location estimates $\hat{G}(q)$ and $\hat{S}$ can be calculated by (2.14) and (2.15) respectively with discretized atomic set $\hat{\mathcal{K}}_d^l$ and coefficients $\Gamma^\star(\hat{\mathcal{K}}_d^l)$, which are the output of Algorithm 2.1.

**Algorithm 2.1** A greedy algorithm for the infinite-dimensional group lasso problem (2.13)

1: **Input:** identification data $(\mathbf{u}^d, \mathbf{y}^d)$, $\varepsilon > 0$, $l_{\max}$
2: Initialize $\hat{\mathcal{K}}_d^0 = \{k_1, k_2, \dots, k_{p_0}\}$.
3: Calculate $\Gamma^\star(\hat{\mathcal{K}}_d^0)$.
4: $l \leftarrow 0$
5: **repeat**
6: Construct a candidate new atom

$$k^+ \leftarrow \operatorname*{argmax}_{k \in \hat{\mathcal{K}}} \left\| \zeta_k^\top R(\hat{\mathcal{K}}_d^l) \right\|_2. \tag{2.20}$$

7: **if** $\left\| \zeta_{k^+}^\top R(\hat{\mathcal{K}}_d^l) \right\|_2 \ge \lambda + \varepsilon$ **then**
8: **begin**
9: $k_{p_0+l+1} \leftarrow k^+$, $\hat{\mathcal{K}}_d^{l+1} \leftarrow \hat{\mathcal{K}}_d^l \cup \{k_{p_0+l+1}\}$
10: Calculate $\Gamma^\star(\hat{\mathcal{K}}_d^{l+1})$ via program (2.13).
11: **end**
12: **else**
13: Break
14: $l \leftarrow l + 1$
15: **until** $l \ge l_{\max}$
16: **Output:** $\hat{\mathcal{K}}_d^l$, $\Gamma^\star(\hat{\mathcal{K}}_d^l)$

Let

$$\hat{\Gamma}^\star := \left\{ \gamma_k^\star \;\middle|\; \gamma_k^\star = \begin{cases} \gamma_i^\star(\hat{\mathcal{K}}_d^l), & k = k_i \in \hat{\mathcal{K}}_d^l \\ \mathbf{0}, & k \in \hat{\mathcal{K}} \setminus \hat{\mathcal{K}}_d^l \end{cases} \right\}. \tag{2.21}$$

Algorithm 2.1 guarantees the following property.

**Proposition 2.1.** *If Algorithm 2.1 terminates without reaching the maximum number of iterations ($l < l_{\max}$), $\hat{\Gamma}^\star$ satisfies the approximate optimality conditions*

$$\begin{cases} \left\|\zeta_k^\top R\right\|_2 < \lambda + \varepsilon, & \text{if } \left\|\gamma_k^\star\right\|_2 = 0, \\ \zeta_k^\top R + \lambda \gamma_k^\star / \left\|\gamma_k^\star\right\|_2 = 0, & \text{if } \left\|\gamma_k^\star\right\|_2 > 0, \end{cases} \tag{2.22}$$

*for all $k \in \hat{\mathcal{K}}$.*

*Proof.* Since $\gamma_k^\star = 0$ for $k \notin \hat{\mathcal{K}}_d^l$ in $\hat{\Gamma}^\star$, we have $R = R(\hat{\mathcal{K}}_d^l)$. For $k \in \hat{\mathcal{K}}_d^l$, the discretized optimality conditions (2.18) guarantee the satisfaction of (2.22). According to Algorithm 2.1, $\left\|\zeta_k^\top R(\hat{\mathcal{K}}_d^l)\right\|_2 = \left\|\zeta_k^\top R\right\|_2 < \lambda + \varepsilon$. So for $k \notin \hat{\mathcal{K}}_d^l$, (2.22) is satisfied since $\left\|\gamma_k^\star\right\|_2 = 0$. □

Proposition 2.1 shows that the infinite-dimensional problem (2.13) is approximately equivalent to the finite-dimensional problem with $(p_0 + l)$ atoms

$$\underset{\{\gamma_i\}_{i=1}^{p_0+l}}{\operatorname{argmin}} \left\| \mathbf{y}^d - \sum_{i=1}^{p_0+l} \zeta_{k_i} \gamma_i \right\|_2^2 + 2\lambda \sum_{i=1}^{p_0+l} \|\gamma_i\|_2 . \tag{2.23}$$

For the rest of the chapter, define $p := p_0 + l$.

The main difficulty in Algorithm 2.1 is solving the non-convex problem (2.20). However, even if (2.20) is not solved exactly, Algorithm 2.1 still guarantees a decrease in the objective function value at each iteration as long as $\left\|\zeta_{k^+}^\top R(\hat{\mathcal{K}}_d^l)\right\|_2 \geq \lambda + \varepsilon$ is satisfied for the candidate atom $k^+$.

### 2.2.3 Debiasing & Pole Location Estimation

Algorithm 2.1 provides a method to solve the group lasso problem (2.13). However, as discussed in Sections 2.1.2 and 2.1.3, solutions to lasso-type regularized problems are known to have a large bias and a large number of false positives in feature selection. To mitigate these problems, the iteratively reweighted adaptive lasso and CPSS are applied to debias the estimate and reject false positives in pole location estimation from Algorithm 2.1.

The iteratively reweighted adaptive approach discussed in Section 2.1.2 is applied to the group lasso problem (2.13) in Algorithm 2.2. It is easy to see that the number of estimated pole locations is non-increasing at each iteration.

CPSS discussed in Section 2.1.3 is applied in Algorithm 2.3. This corresponds to replacing the loss function

$$\mathcal{L}(\cdot) = \left\| \mathbf{y}^d - \sum_{i=1}^{p} \zeta_{k_i} \gamma_i \right\|_2^2 \tag{2.25}$$

**Algorithm 2.2** Iteratively reweighted adaptive group lasso

1: **Input:** identification data $(\mathbf{u}^d, \mathbf{y}^d)$, $\varepsilon > 0$, $m_s$
2: Find $\hat{\mathcal{K}}_d^l = \{k_1, \ldots, k_p\}$, $\Gamma^\star(\hat{\mathcal{K}}_d^l) := \left\{\gamma_1^{\star,0}, \ldots, \gamma_p^{\star,0}\right\}$ from Algorithm 2.1.
3: **for** $m = 1$ **to** $m_s$ **do**
4: **begin**
5: Find $\left\{\gamma_i^{\star,m}\right\}_{i=1}^p$ by solving

$$\underset{\{\gamma_i\}_{i=1}^p}{\operatorname{argmin}} \left\| \mathbf{y}^d - \sum_{i=1}^p \zeta_{k_i}\gamma_i \right\|_2^2 + 2\lambda \sum_{i=1}^p \frac{\|\gamma_i\|_2}{\left\|\gamma_i^{\star,m-1}\right\|_2 + \varepsilon}. \tag{2.24}$$

6: **end**
7: Calculate $\hat{G}(q)$ by (2.14) with discretized atomic set $\hat{\mathcal{K}}_d^l$ and coefficients $\left\{\gamma_i^{\star,m_s}\right\}_{i=1}^p$.
8: **Output:** $\hat{G}(q)$

with

$$\mathcal{L}_B(\cdot) := \left\| \mathbf{y}^d(B) - \sum_{i=1}^p \zeta_{k_i}(B,:)\gamma_i \right\|_2^2, \tag{2.26}$$

where $B \subset \{1,2,\ldots,N\}$ defines a random subsample of data. The transfer function can also be estimated by least squares on the stable solution of the atomic set.

**Algorithm 2.3** Complementary pairs stability selection (CPSS)

1: **Input:** identification data $(\mathbf{u}^d, \mathbf{y}^d)$, $\tau \in (0.5, 1]$, $n_s$
2: Find $\hat{\mathcal{K}}_d^l$ from Algorithm 2.1.
3: **for** $i = 1$ **to** $n_s$ **do**
4: **begin**
5: Generate a random subsample $B_i \subset \{1,2,\ldots,N\}$ with $\lfloor N/2 \rfloor$ elements.
6: $\bar{B}_i \leftarrow \{1,2,\ldots,N\} \setminus B_i$
7: Calculate $\hat{S}_{B_i}$, $\hat{S}_{\bar{B}_i}$ by solving (2.23) with the loss function $\mathcal{L}(\cdot)$ replaced by $\mathcal{L}_{B_i}(\cdot)$, $\mathcal{L}_{\bar{B}_i}(\cdot)$ respectively.
8: **end**
9: $\hat{S} \leftarrow \left\{k \,\middle|\, \frac{1}{2n_s}\sum_{i=1}^{n_s}\left(\mathbb{1}_{\hat{S}_{B_i}}(k) + \mathbb{1}_{\hat{S}_{\bar{B}_i}}(k)\right) \geq \tau\right\}$, where $\mathbb{1}$ denotes the indicator function.
10: **Output:** $\hat{S}$

### 2.2.4 Numerical Results

The performances of the proposed algorithms are assessed by numerical simulation on a benchmark fourth-order system previously analyzed in Landau et al. (1995):

$$G_1(q) = \frac{0.10884q + 0.19513}{q^4 - 1.41833q^3 + 1.58939q^2 - 1.31608q + 0.88642}. \tag{2.27}$$

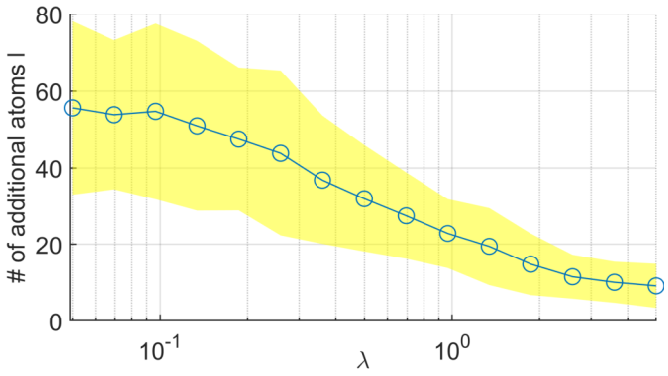



Figure 2.4: The number of additional atoms $l$ in Algorithm 2.1 for $\sigma^2 = 0.1$. Blue: mean values, yellow: ranges within one standard deviation.

The system has been normalized to have an $\mathcal{H}_2$-norm of 1. In what follows, results obtained with Algorithms 2.1, 2.2, and 2.3 are labelled by *InfA*, *AdpInfA*, and *SS* respectively.

Identification data of length $N = 100$ are generated with zero-mean i.i.d. unit Gaussian inputs from a zero initial condition. Two noise levels $\sigma^2 = 0.1$ and 0.01 are considered. The atomic responses $\phi_k$ are also generated from a zero initial condition. 100 Monte Carlo simulations are conducted for each noise level. The initial discretized atomic set $\hat{\mathcal{K}}_d^0$ contains $p_0 = 50$ randomly generated atoms with $k_i = \alpha_i \exp(j\beta_i)$, where $\alpha_i$ and $\beta_i$ are subject to uniform distributions in $[0, 1)$ and $[0, \pi]$ respectively. Finite-dimensional group lasso problems are solved by MOSEK. The candidate atom generation problem (2.20) is solved by the particle swarm solver in MATLAB. The hyperparameter $\lambda$ is selected by cross-validation from a 15-point log-space grid between 0.05 and 5 for $\sigma^2 = 0.1$ and between 0.005 and 0.5 for $\sigma^2 = 0.01$, except for *SS* where $\lambda$ is fixed to 0.5 for $\sigma^2 = 0.1$ and 0.05 for $\sigma^2 = 0.01$. The following parameters are used in simulation: $\varepsilon = 10^{-5}$, $\tau = 0.9$, $n_s = 50$, and $m_s = 2$.

First, the number of additional atoms $l$ required in Algorithm 2.1 is plotted against the $\lambda$ values in Figure 2.4. The maximum $l$ in all Monte Carlo simulations is 118, which is below the $l_{\max}$ setting. Results show that the proposed greedy atom generation approach can converge within a reasonable number of iterations, and the required number of additional atoms decreases with $\lambda$.

To demonstrate the performance of the proposed algorithms, they are compared to three benchmark algorithms: 1) least-squares estimation with an ARX model and a known true model order (*ARX*); 2) kernel-based identification with a TC kernel design (*TCK*) (Chen et al., 2012), where hyperparameters are obtained by the empirical Bayes approach; 3) discretized atomic norm regularization in Shah et al. (2012) with 50 (*Atom*) and 500 (*Atom2*) random atoms. Note that *Atom2* uses a significantly larger atomic set compared to Algorithm 2.1, as shown in Figure 2.4.

Figure 2.5 compares the identification accuracy of all algorithms in terms of the impulse response fitting $W$. It can be seen that the three proposed algorithms all perform better than the benchmark algorithms at both noise levels. In particular, *InfA* obtains better fitting compared to *Atom2*,

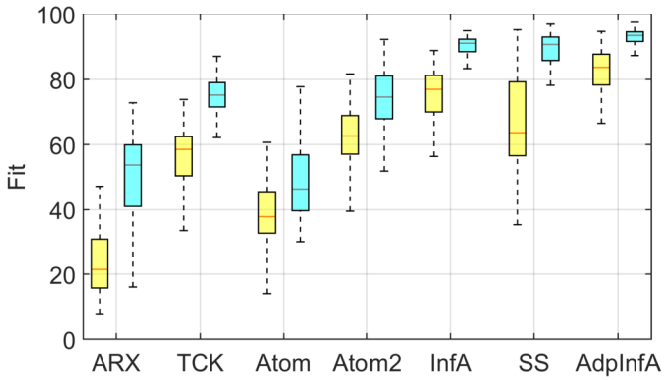



Figure 2.5: Boxplot of impulse response fitting. Yellow: $\sigma^2 = 0.1$, cyan: $\sigma^2 = 0.01$.

Table 2.1: Bias-variance analysis of impulse response estimation.

| | **TCK** | **Atom** | **Atom2** | **InfA** | **SS** | **AdpInfA** |
|---|---|---|---|---|---|---|
| $\sigma^2 = 0.1$ | | | | | | |
| Bias$^2$ $[\times 10^{-2}]$ | 6.76 | 23.42 | 6.34 | 2.63 | 8.28 | 0.91 |
| Var $[\times 10^{-2}]$ | 13.04 | 13.59 | 8.52 | 3.80 | 15.68 | 2.70 |
| MSE $[\times 10^{-2}]$ | 19.80 | 37.01 | 14.86 | 6.44 | 23.96 | 3.60 |
| $\sigma^2 = 0.01$ | | | | | | |
| Bias$^2$ $[\times 10^{-2}]$ | 1.78 | 15.92 | 2.22 | 0.43 | 0.47 | 0.07 |
| Var $[\times 10^{-2}]$ | 5.45 | 11.68 | 5.26 | 0.76 | 3.12 | 0.52 |
| MSE $[\times 10^{-2}]$ | 7.23 | 27.60 | 7.48 | 1.18 | 3.59 | 0.59 |

which uses a much larger atomic set. This demonstrates the effectiveness of the proposed atom generation approach. *AdpInfA* further improves on the identification accuracy of *InfA* with iterative reweighting.

To further investigate the sources of the estimation errors, Table 2.1 shows the bias-variance analysis of impulse response estimation. As an algorithm proposed to debias the estimate, *AdpInfA* indeed produces a much smaller bias than all other algorithms. This is also the main contributor to the reduction of mean squared error (MSE) compared to the baseline *InfA* algorithm, on which *AdpInfA* is based.

Finally, the capability of estimating the poles of the system is demonstrated in Figures 2.6 and 2.7. It is illustrated in Figure 2.6 that all the algorithms that directly solve group lasso problems estimate a much larger number of poles compared to the true one, similar to the phenomenon observed in Example 2.2. *AdpInfA* mitigates the over-estimation since the active atomic set shrinks at each iteration, whereas SS obtains a very accurate model order estimation.

To assess the accuracy of pole location estimation, Figure 2.7 further compares the distributions of estimated pole locations in all 100 Monte Carlo simulations. Despite knowing the true model order, *ARX* fails to estimate the pole locations accurately. Although the estimated model order

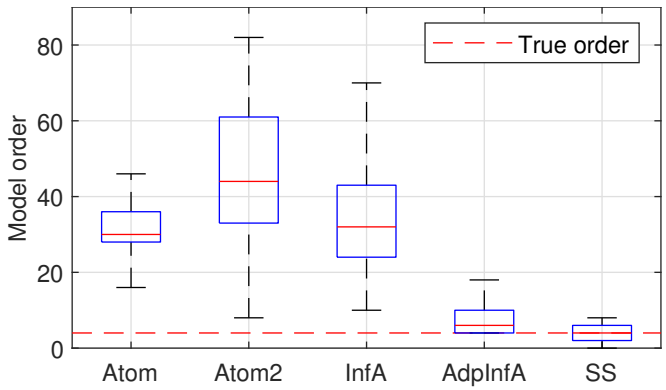



Figure 2.6: Comparison of estimated model orders for $\sigma^2 = 0.1$.

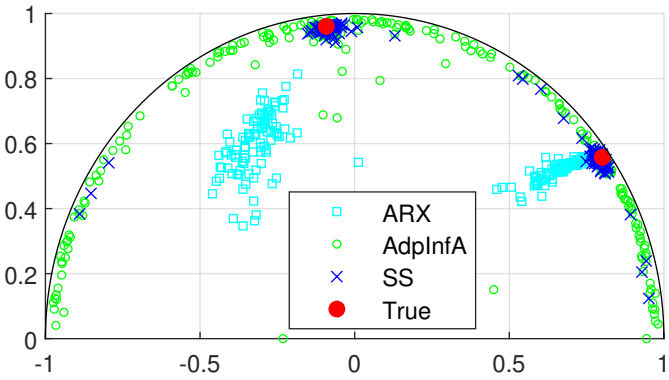



Figure 2.7: Distributions of estimated pole locations in all 100 Monte Carlo simulations for $\sigma^2 = 0.1$.

is close to the true one, *AdpInfA* estimates a significant number of false positives in terms of the actual pole locations. Among all the algorithms, only *SS* can obtain accurate pole location estimations with few false positives, proving the effectiveness of the CPSS method.

## 2.3 Summary

This chapter applies advanced techniques studied in high-dimensional statistics to the atomic norm regularization problem in linear system identification. A greedy algorithm is presented to generate new candidate atomic models from infinitely many possible pole locations. Common drawbacks of lasso-type regularization are mitigated by adaptively adjusting the regularization weights for each atom and selecting only repeatedly occurring pole locations from subsamples of data.

Results in this chapter suggest that sparse learning algorithms are a promising alternative to kernel-based methods with fewer design requirements and direct pole location estimation. Further research directions include improvements in computational efficiency, comparison with model

order reduction methods, and extensions to multiple-input multiple-output systems and frequency-domain data.

# 3 Kernel Learning in System Identification

In this chapter, we first introduce the regularized impulse response estimator with three different interpretations in Section 3.1. Then, the critical problem of designing the weighting matrix $K$ in the estimator is discussed in Section 3.2.

Section 3.3 proposes a novel multiple regularization method that promotes low-complexity model structure in terms of the number of poles. Multiple regularization has been applied with tuned/correlated (TC) kernels (Chen et al., 2014; Hong et al., 2018) and filters (Chen et al., 2018) with successful applications to complex systems. The novelty of the proposed method is in both the design of basis regularization matrices and the hyperparameter tuning. The basis regularization matrices are designed to be optimal regularization matrices for first-order systems. This design shows that the number of poles in the identified model corresponds to the cardinality of the hyperparameters. Then, the hyperparameters are estimated using a maximum *a posteriori* (MAP) approach. By selecting a sparse hyperprior for the hyperparameters, the MAP hyperparameter tuning gives a sparse estimation of the hyperparameters. The resulting optimization problem has the form of difference of convex programming (DCP) problems, which can be efficiently solved. Simulation results demonstrate that the proposed method achieves a better bias-variance trade-off and a better fit to the model than existing methods.

Error-bound quantification has also been an important topic in kernel and GP learning. In kernel-based linear system identification, Pillonetto and Scampicchio (2022) establishes non-asymptotic bounds for all stable systems with bounded pole magnitudes. However, the bounds are too conservative and thus only useful for sample complexity analysis. Error bounds are also widely studied in GP literature, e.g., Maddalena et al. (2021); Srinivas et al. (2012), usually obtained by scaling posterior standard deviations. Such bounds are derived assuming the prior covariance function is exact and/or an upper bound of the RKHS-induced norm is known. However, both assumptions are impractical in general. Several works provide modified bounds considering the discrepancy between the applied and the true kernel functions (Capone et al., 2022; Beckers et al., 2018; Fiedler and Lucia, 2021; Tuo and Wang, 2022). These results depend on knowledge of the magnitude of the discrepancy, which is usually not known *a priori*. Such information is estimated

from data in Capone et al. (2022) by investigating the maximum marginal likelihood problem in hyperparameter estimation. Unfortunately, these works all consider an identity regressor, which is not common in system identification, and often consider kernel classes that do not contain the typical stable kernels used in linear system identification. Sampling-based approaches have also been proposed. The sign-perturbed sums approach is used in Baggio et al. (2022) by randomly perturbing the sign of model residuals. The Markov chain Monte Carlo approach is used in Pillonetto and Ljung (2023) to approximate the full posterior distribution. However, such bounds are based on sampling and thus do not admit an easy-to-use analytic form.

Section 3.4 provides probabilistic error bounds for kernel-based linear system identification with no prior knowledge of the hyperparameters by extending Capone et al. (2022) to general regression problems and stable kernels. The proof in Capone et al. (2022) is also simplified with an improved constant. Our approach assumes the correct kernel structure and a known hyperprior that describes the distribution of the hyperparameters. A high-probability set is first estimated for the hyperparameters from the marginal likelihood function. Then, the worst-case posterior covariance is found within the range of hyperparameters. A uniform bound is obtained for diagonal and tuned/correlated (TC) kernels. For general kernels, element-wise bounds can be found by optimization. Finally, probabilistic error bounds are established by scaling the worst-case posterior standard deviations. Optimization problems to obtain the tightest error bounds are discussed as well. Numerical simulations demonstrate that the proposed error bounds can provide high-probability bounds of the estimation error in practice.

## 3.1 The Threefold Interpretation of Kernel-Based Identification

As mentioned in Section 1.5, in this chapter, the problem of identifying the impulse response $(g_l)_{l=0}^{\infty}$ is considered. The stability of $G_0(q)$ implies that the impulse response decays exponentially. Thus, it is reasonable to truncate the infinite impulse response at a sufficiently high order, denoted by $n_g$. Accordingly, the system is approximated with a finite-length impulse response of $\mathbf{g} := [g_0, g_1, \ldots, g_{n_g-1}]^\mathsf{T} \in \mathbb{R}^{n_g}$. This leads to the finite impulse response (FIR) model of the system, which is defined as $G_0(q) = \sum_{l=0}^{n_g-1} g_l q^{-1}$, i.e.,

$$y_t = \sum_{l=0}^{n_g-1} g_l u_{t-l} + v_t. \tag{3.1}$$

Using the identification data $(\mathbf{u}^d, \mathbf{y}^d)$, the following data equation is constructed:

$$\underbrace{\begin{bmatrix} y_1^d \\ y_2^d \\ \vdots \\ y_N^d \end{bmatrix}}_{\mathbf{y}^d} = \underbrace{\begin{bmatrix} u_1^d & 0 & \cdots & 0 \\ u_2^d & u_1^d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ u_{N-1}^d & u_{N-2}^d & \cdots & u_{N-n_g}^d \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n_g-1} \end{bmatrix}}_{\mathbf{g}} + \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}}_{\mathbf{v}}, \tag{3.2}$$

where, for the sake of simplicity, a zero initial condition is assumed, i.e., $u_t^d = 0, \forall t \leq 0$. This is used as a regression model for estimating $\mathbf{g}$.

Since $v_t \sim \mathcal{N}(0, \sigma^2)$, we have $p(\mathbf{y}^d|\mathbf{u}^d, \mathbf{g}) \sim \mathcal{N}(\Phi\mathbf{g}, \sigma^2\mathbb{I})$. If no prior knowledge is assumed for $G_0(q)$, the maximum likelihood estimator of $\mathbf{g}$ is given by the least-squares solution

$$\hat{\mathbf{g}}^{\mathrm{LS}} := \underset{\mathbf{g}}{\operatorname{argmax}}\ p(\mathbf{y}^d|\mathbf{u}^d, \mathbf{g}) = \underset{\mathbf{g}}{\operatorname{argmin}}\ \left\|\mathbf{y}^d - \Phi\mathbf{g}\right\|_2^2 = \left(\Phi^\top\Phi\right)^{-1}\Phi^\top\mathbf{y}^d, \tag{3.3}$$

where $\Phi$ and $\mathbf{g}$ are defined in (3.2). The least-squares solution is the best unbiased estimator with i.i.d. Gaussian output noise (Ljung, 1999). It is well-known that the estimation error is also Gaussian with covariance $\operatorname{cov}(\mathbf{g}) = \sigma^2\left(\Phi^\top\Phi\right)^{-1} =: \Sigma^{\mathrm{LS}}$. Element-wise stochastic error bounds can be obtained for $\hat{\mathbf{g}}^{\mathrm{LS}}$ as

$$\mathbb{P}\left(\left|\hat{g}_l^{\mathrm{LS}} - g_l\right| \leq \mu_\delta\sqrt{\Sigma_{l,l}^{\mathrm{LS}}}\right) \geq 1 - \delta, \tag{3.4}$$

where $\mu_\delta$ is the two-tailed quantile function of the Gaussian distribution, given by

$$F_{\mathcal{N}}(\mu_\delta) \geq 1 - \delta/2, \tag{3.5}$$

$F_{\mathcal{N}}(\cdot)$ is the cumulative distribution function of the Gaussian distribution.

However, since the parameter dimension $n_g$ is typically large, the estimate's variance can be high under high noise levels, a phenomenon known as overfitting. This issue can be alleviated by introducing suitable regularizers into (3.3) (Pillonetto et al., 2014; Chen, 2018). Note that the impulse response of the stable system $G_0(q)$ is typically smooth and exponentially converges to zero. Such prior knowledge can be encoded as 1) a basis decomposition, 2) a prior distribution in GP regression, or 3) an RKHS in kernel regression. In all three cases, the nominal estimate of $\mathbf{g}$ is given by the regularized least-squares solution

$$\hat{\mathbf{g}} := \underset{\mathbf{g}}{\operatorname{argmin}}\ \left\|\mathbf{y}^d - \Phi\mathbf{g}\right\|_2^2 + \sigma^2\mathbf{g}^\top K^{-1}\mathbf{g} = \left(\Phi^\top\Phi + \sigma^2K^{-1}\right)^{-1}\Phi^\top\mathbf{y}^d. \tag{3.6}$$

With different interpretations of $K$, the regularized estimate $\hat{\mathbf{g}}$ has a threefold meaning.

**Ridge regression with basis decomposition.** In the basis decomposition interpretation, the impulse response vector $\mathbf{g}$ is parametrized by bases $(\mathbf{g}_i)_{i=1}^{n_b}$, i.e., $\mathbf{g} = \sum_{i=1}^{n_b}\alpha_i\mathbf{g}_i = G\alpha$, where $n_b$ is the dimension of the bases, $G := [\mathbf{g}_1\ \mathbf{g}_2\ \ldots\ \mathbf{g}_{n_b}] \in \mathbb{R}^{n_g\times n_b}$ collects all the bases, and $\alpha := \operatorname{col}(\alpha_1, \alpha_2, \ldots, \alpha_{n_b})$ collects all the coefficients. The coefficients can be estimated by the following ridge regression problem:

$$\hat{\alpha} := \underset{\alpha}{\operatorname{argmin}}\ \left\|\mathbf{y}^d - \Phi G\alpha\right\|_2^2 + \sigma^2\left\|\alpha\right\|_2^2 = \left(G^\top\Phi^\top\Phi G + \sigma^2\mathbb{I}\right)^{-1}G^\top\Phi^\top\mathbf{y}^d. \tag{3.7}$$

By selecting $K = GG^\top$, we have

$$\begin{aligned}\hat{\mathbf{g}} &= \left(\Phi^\top\Phi + \sigma^2 K^{-1}\right)^{-1}\Phi^\top \mathbf{y}^d = \left(GG^\top\Phi^\top\Phi + \sigma^2\mathbb{I}\right)^{-1} GG^\top\Phi^\top\mathbf{y}^d \\ &= G\left(G^\top\Phi^\top\Phi G + \sigma^2\mathbb{I}\right)^{-1} G^\top\Phi^\top\mathbf{y}^d = G\hat{\alpha}.\end{aligned} \tag{3.8}$$

This means that $K$ can be interpreted as the sum of the outer products of the basis impulse responses.

**Maximum *a posteriori* estimate with a Gaussian prior.** In the GP regression interpretation, $K$ is selected as the covariance of the prior distribution of $\mathbf{g}$: $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, K)$. Then $\mathbf{y}^d$ and $\mathbf{g}$ are jointly Gaussian:

$$\begin{bmatrix}\mathbf{g} \\ \mathbf{y}^d\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}K & K\Phi^\top \\ \Phi K & \Phi K\Phi^\top + \sigma^2\mathbb{I}\end{bmatrix}\right). \tag{3.9}$$

From the property of Gaussian distribution, the distribution of $\mathbf{g}$ given $\mathbf{y}^d$ is also Gaussian: $\mathbf{g}|\mathbf{u}^d, \mathbf{y}^d \sim \mathcal{N}(\hat{\mathbf{g}}, \Sigma)$, where the posterior mean is the estimate $\hat{\mathbf{g}}$ and $\Sigma := \sigma^2\left(\Phi^\top\Phi + \sigma^2 K^{-1}\right)^{-1}$ is the posterior covariance (Chen et al., 2012). This means that the MAP estimate of $\mathbf{g}$ given the prior distribution is given by $\hat{\mathbf{g}}$: $\hat{\mathbf{g}} = \underset{\mathbf{g}}{\operatorname{argmax}}\ p(\mathbf{g}|\mathbf{u}^d, \mathbf{y}^d)$.

From the posterior covariance, the associated element-wise stochastic error bounds are

$$\mathbb{P}\left(|\hat{g}_l - g_l| \le \mu_\delta\sqrt{\Sigma_{l,l}}\right) \ge 1 - \delta, \tag{3.10}$$

conditioned on the identification data. The bounds assume a random design of $\mathbf{g}$ and that the prior distribution of $\mathbf{g}$ is correct.

**Regularization in a reproducing kernel Hilbert space.** In the kernel regression interpretation (Saitoh and Sawano, 2016, Chapter 3), the *continuous-time* impulse response function $g(t) : [0, +\infty) \to \mathbb{R}$, $g(l) = g_l$, $l = 0, \ldots, n_g - 1$ is identified by solving the regularized function learning problem within an RKHS $\mathcal{H}$ associated with a kernel function $k(\cdot,\cdot) : [0,+\infty) \times [0,+\infty) \to \mathbb{R}$:

$$\begin{aligned}g^\star(\cdot) := \arg\min_{g(\cdot)\in\mathcal{H}}\ &\left\|\mathbf{y}^d - \Phi\mathbf{g}\right\|_2^2 + \sigma^2\left\|g(\cdot)\right\|_{\mathcal{H}}^2 \\ &\text{s.t. } \mathbf{g} = \begin{bmatrix}g(0) & \ldots & g(n_g-1)\end{bmatrix}^\top,\end{aligned} \tag{3.11}$$

where the regularizer $\|g(\cdot)\|_{\mathcal{H}}$ is the norm of $g(\cdot)$ induced by $\mathcal{H}$.

From the representer theorem (Schölkopf et al., 2001), the optimal continuous-time impulse response function for (3.11) is given by $g^\star(x) = \mathbf{k}_x\left(\Phi^\top\Phi K + \sigma^2\mathbb{I}\right)^{-1}\Phi^\top\mathbf{y}^d$, where $K$ evaluates the kernel function associated with the RKHS $\mathcal{H}$ at $l = 0, \ldots, n_g - 1$, i.e., $K_{l,l} = k(l,l)$, and $\mathbf{k}_x := [k(x,0)\ \ldots\ k(x,n_g-1)]$. The corresponding optimal discrete-time impulse response vector is $\mathbf{g}^\star = \hat{\mathbf{g}}$. The induced norm of $g^\star$ is calculated as $\|g^\star(\cdot)\|_{\mathcal{H}}^2 = \hat{\mathbf{g}}^\top K^{-1}\hat{\mathbf{g}}$. This means that $\hat{\mathbf{g}}$ is the discrete-time restriction of the solution to the regularized function learning problem when $K$

evaluates the kernel function values at discrete time points.

## 3.2 Kernel Design and Hyperparameter Selection

As seen from the previous section, the regularization weighting matrix $K$ is critical to the performance of the kernel-based method. Extensive studies have been conducted to obtain appropriate structures of $K$ that promote impulse response estimates that are both smooth and exponentially converge to zero (Chen, 2018). These structures parametrize the kernel with hyperparameters $\eta \in \mathcal{E} \subseteq \mathbb{R}^{n_\eta}$: $K := K(\eta)$.

The hyperparameters must be selected before applying the estimator (3.6). The most widely used approach to hyperparameter selection is the maximum marginal likelihood method, also known as the empirical Bayes method. It uses the GP regression interpretation and maximizes the probability of observing $\mathbf{y}^d$ given the inputs $\mathbf{u}^d$ and the hyperparameters $\eta$:

$$\hat{\eta} := \underset{\eta}{\operatorname{argmin}} \; -\log p(\mathbf{y}^d|\mathbf{u}^d, \eta), \tag{3.12}$$

where

$$p(\mathbf{y}^d|\mathbf{u}^d, \eta) := \exp\left(-\frac{1}{2}\log\det\Psi(\eta) - \frac{1}{2}\left(\mathbf{y}^d\right)^\top \Psi^{-1}(\eta)\mathbf{y}^d + \text{const.}\right) \tag{3.13}$$

and $\Psi(\eta) := \sigma^2\mathbb{I} + \Phi K(\eta)\Phi^\top$. If the prior distribution of the hyperparameters $p(\eta)$, known as the *hyperprior*, is available, $\eta$ can also be estimated using an MAP approach:

$$\hat{\eta}^{\text{MAP}} := \underset{\eta}{\operatorname{argmin}} \; -\log p(\mathbf{y}^d|\mathbf{u}^d, \eta)p(\eta). \tag{3.14}$$

The estimated hyperparameters $\hat{\eta}$ are used, with certainty equivalence, to construct $K(\hat{\eta})$ and then to obtain the estimate $\hat{\mathbf{g}}$.

Dedicated kernel structures have been designed for linear system identification. The most commonly used ones include:

1. diagonal (DI): $K^{\text{DI}}_{i,i}(\eta) = c_K\lambda_K^i$, $K^{\text{DI}}_{i,j}(\eta) = 0$ for $i \neq j$,
2. tuned/correlated (TC): $K^{\text{TC}}_{i,j}(\eta) = c_K\lambda_K^{\max(i,j)}$,
3. stable spline (SS): $K^{\text{SS}}_{i,j}(\eta) = c_K\lambda_K^{2\max(i,j)}\left(\frac{\lambda_K^{\min(i,j)}}{2} - \frac{\lambda_K^{\max(i,j)}}{6}\right)$,

where $\eta := [c_K \;\; \lambda_K]^\top \in \mathcal{E} := \left\{[c_K \;\; \lambda_K]^\top \middle| \, c_K \geq 0, 0 \leq \lambda_K \leq 1\right\}$ are the hyperparameters. These kernel designs have been shown effective both theoretically and numerically (Pillonetto et al., 2022).

Alternatively, $K$ can be parametrized as a linear combination of a family of basis matrices $K_i$, i.e., $K(\eta) = \sum_{i=1}^{n_\eta} \eta_i K_i$, for all $\eta \in \mathcal{E}$. This structure is known as multiple kernel design in Chen et al. (2014). When basis matrices $K_i$ have a rank of 1, they can be interpreted as outer products of

basis impulse responses as discussed in Section 3.1. Section 3.3 investigates the following two main problems in multiple kernel design.

*What is the appropriate choice of $K_i$?* A suitable structure of basis matrices $K_i$ is proposed for estimating a model with control of the number of poles.

*What is the appropriate approach to tune $\eta_i$?* In contrast to conventional single kernel design, where hyperparameter tuning is often conducted by non-convex optimization or grid search since $n_\eta$ is small, a high-dimensional hyperparameter tuning problem must be addressed in multiple kernel design. The high-dimensional problem is then prone to high variance and computational intractability. A sparse hyperparameter tuning approach is presented using MAP estimation with a sparse hyperprior.

Given a well-tuned weighting matrix $K$, the kernel-based method leads to a nominal estimate with the desired model structure and a satisfactory bias-variance trade-off. However, the effect of hyperparameter estimation on the error bounds (3.10) is unknown. In Section 3.4, we construct error bounds when the hyperparameters are estimated with a single kernel design.

## 3.3 Sparse Kernel Design in Impulse Response Estimation

### 3.3.1 From Model Complexity to Hyperparameter Sparsity

To reveal the number of poles in the model, we again start from the atomic decomposition (1.5) used in Chapter 2. Define the atomic impulse response $\mathbf{g}_k$ as the finite impulse response of $A_k(q)$ given by

$$\mathbf{g}_k := \left(1-|k|^2\right)[0\ 1\ k\ \ldots\ k^{n_g-2}]^{\mathsf{T}} \in \mathbb{C}^{n_g}, \quad \mathbf{g}_1 := [1\ 0\ \ldots\ 0]^{\top} \tag{3.15}$$

and the atomic weighting matrix

$$K_k := \mathbf{g}_k \left(\mathbf{g}_k\right)^{\mathsf{H}}. \tag{3.16}$$

As proved in Chen et al. (2012), when the underlying system is exactly $A_k(q)$, the optimal weighting matrix is given by $K_k$. As discussed in Sections 3.1, a summation of $K_k$ corresponds to a basis impulse response design of $\mathbf{g}_k$. Both facts motivate the selection of $K_k$ as the basis matrix in multiple kernel design.

Therefore, the following structure of $K(\eta)$ is proposed:

$$K(\eta) = \sum_{i=1}^{n_\eta} \eta_i K_{k_i} = \sum_{i=1}^{n_\eta} \eta_i \mathbf{g}_{k_i} \left(\mathbf{g}_{k_i}\right)^{\mathsf{H}}, \tag{3.17}$$

where $\left\{k_1, k_2, \ldots, k_{n_\eta}\right\} =: \mathcal{K}_d \subset \mathcal{K}$ are a dense discretization of all the stable poles $\mathcal{K}$ and the hyperparameters $\left[\eta_1\ \eta_2\ \ldots\ \eta_{n_\eta}\right]^{\mathsf{T}} \in \mathbb{R}_+^{n_\eta}$ are denoted by $\eta$. For real-valued systems, it is required that for any $k \in \mathcal{K}_d$, one has $\bar{k} \in \mathcal{K}_d$, where the overbar denotes the complex conjugate. Furthermore, the weighting matrix $K_\eta$ should also be real-valued. Noting that $K_{k_i} = \bar{K}_{k_j}$ for any

$k_i = \bar{k}_j$, $\eta$ needs to be constrained to satisfy

$$\eta_i = \eta_j, \quad \forall i, j \text{ such that } k_i = \bar{k}_j. \tag{3.18}$$

In Section 3.1, regularized estimate (3.6) assumes a positive definite $K$ with a well-defined matrix inverse. However, multiple kernel design (3.17) might only be positive semi-definite since it is a summation of rank-1 positive semi-definite matrices. A straightforward extension to (3.6) for a positive semi-definite $K$ is given as follows.

Let the singular value decomposition (SVD) of $K$ be given as

$$K = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} K' & \mathbb{0} \\ \mathbb{0} & \mathbb{0} \end{bmatrix} \begin{bmatrix} U_1^\mathsf{T} \\ U_2^\mathsf{T} \end{bmatrix}, \tag{3.19}$$

where $K'$ is a diagonal matrix with positive diagonal entries. The regularized impulse response estimate $\hat{\mathbf{g}}$ is defined as the solution of the following constrained regularized problem (Chen et al., 2014):

$$\hat{\mathbf{g}} = \underset{\mathbf{g}}{\operatorname{argmin}} \ \|\mathbf{y}^d - \Phi\mathbf{g}\|^2 + \sigma^2 \mathbf{g}^\mathsf{T} U_1 \left(K'\right)^{-1} U_1^\mathsf{T} \mathbf{g} \qquad \text{s.t.} \quad U_2^\mathsf{T}\mathbf{g} = 0 \tag{3.20}$$

$$= K\Phi^\top \left(\Phi K \Phi^\top + \sigma^2 \mathbb{I}\right)^{-1} \mathbf{y}^d \tag{3.21}$$

Problem (3.20) constraints the estimate to the range of $U_1$. Note that

$$\operatorname{range}(U_1) = \operatorname{span}\left(\{\mathbf{g}_{k_i}\}_{\eta_i \neq 0}\right). \tag{3.22}$$

This leads to the following lemma.

**Lemma 3.1.** *Let $K = K(\eta)$ given in* (3.17)*. There exists $\mathbf{c} := [c_1 \ c_2 \ \dots \ c_{n_\eta}]^\mathsf{T} \in \mathbb{C}^{n_\eta}$ such that the regularized estimate $\hat{\mathbf{g}}$ in* (3.21) *can be decomposed as $\hat{\mathbf{g}} = \sum_{i=1}^{n_\eta} c_i \mathbf{g}_{k_i}$, where $c_i = 0$ for all $\eta_i = 0$.*

According to Lemma 3.1, the desired low-complexity structure of the estimated impulse response is induced by the sparsity of the hyperparameters $\eta$. However, to have a favorable impulse response estimation, the set $\mathcal{K}_d$ should be nearly dense in $\mathcal{K}$. Accordingly, employing a large set of atomic weighting matrices is advantageous, i.e., $n_\eta$ should be large. This suggests performing sparse estimation at the level of hyperparameter tuning, which will be discussed in the following subsection. Meanwhile, a satisfactory bias-variance trade-off at the level of impulse response estimation is maintained by utilizing the regularized identification method in (3.20).

**Remark 3.1.** *This approach differs from the atomic norm regularization discussed in Chapter 2, where $l_1$-norm regularization is directly imposed on the decomposition coefficients $\mathbf{c}$ at the level of impulse response estimation. This direct sparsity regularization adds more regularization to*

*the impulse response estimation. The atomic norm regularization is known to suffer from high bias in its basic form, as shown in Section 2.1.2.*

### 3.3.2 Maximum *a Posteriori* Estimation of Hyperparameters

The empirical Bayes method (3.12) performs well when the number of hyperparameters is small, and the data set is not too small or noisy. However, as discussed in the previous subsection, a considerably large number of hyperparameters are to be estimated here. In this situation, the empirical Bayes method is prone to high variance. Therefore, employing the MAP approach is preferable, especially when prior knowledge of the hyperparameter sparsity is available.

To impose additional sparsity regularization for estimating the hyperparameters introduced in the previous subsection, the MAP approach (3.14) is used with a hyperprior that induces sparsity.

Define set $\mathcal{K}_d^+ := \{k_i \in \mathcal{K}_d \,|\, \Im(k_i) \geq 0\}$. According to the structural constraint of $K(\eta)$ given in (3.18), we only need to estimate $\eta_i$ for $k_i \in \mathcal{K}_d^+$. The remaining hyperparameters are determined automatically. In this subsection, with an abuse of notation, $\eta$ denotes the vector of independent hyperparameters $(\eta_i)_{k_i \in \mathcal{K}_d^+}$ and $n_\eta$ denotes the cardinality of $\mathcal{K}_d^+$.

The hyperprior for $\eta_i$ is selected as an i.i.d. exponential distribution (Aravkin et al., 2014). More precisely, we have $p(\eta_i) := \lambda_\eta \exp(-\lambda_\eta \eta_i)$ with support $\eta_i \geq 0$, where $\lambda_\eta > 0$ is the *rate parameter* of the distribution, which can be considered as the "hyper-hyperparameter" that parametrizes the hyperprior. In this paper, $\lambda_\eta$ is selected by cross-validation. The prior distribution of $\eta$ is thus given by

$$p(\eta) = \lambda_\eta^{n_\eta} \exp\left(-\lambda_\eta \sum_{i=1}^{n_\eta} \eta_i\right), \quad \eta \geq \mathbf{0}. \tag{3.23}$$

Substituting (3.23) and (3.13) into (3.14), we have

$$\hat{\eta}^{\mathrm{MAP}} = \underset{\eta}{\operatorname{argmin}} \underbrace{\frac{1}{2}\log\det\Psi(\eta) + \frac{1}{2}\left(\mathbf{y}^d\right)^\top \Psi^{-1}(\eta)\mathbf{y}^d + \lambda_\eta \sum_{i=1}^{n_\eta} \eta_i}_{J_{\mathrm{REB}}(\eta)}, \qquad \text{s.t.} \quad \eta \geq \mathbf{0}. \tag{3.24}$$

Note that $J_{\mathrm{REB}}(\eta) = J_{\mathrm{REB}}^1(\eta) - J_{\mathrm{REB}}^2(\eta)$, where

$$J_{\mathrm{REB}}^1(\eta) = \frac{1}{2}\left(\mathbf{y}^d\right)^\top \Psi^{-1}(\eta)\mathbf{y}^d + \lambda_\eta \sum_{i=1}^{n_\eta} \eta_i, \quad J_{\mathrm{REB}}^2(\eta) = -\frac{1}{2}\log\det\Psi(\eta) \tag{3.25}$$

are both convex functions. So optimization problem (3.24) can be expressed as a DCP problem, which can be solved efficiently with existing algorithms such as Shen et al. (2016); Yuille and Rangarajan (2002).

The difference between (3.24) and the empirical Bayes method (3.12) is the $\lambda_\eta \sum_{i=1}^{n_\eta} \eta_i$ term in the objective function, which comes from the sparse hyperprior. This term performs regularization on hyperparameter estimation to improve the bias-variance trade-off. Therefore, the MAP estimate (3.24) can be alternatively called the *regularized empirical Bayes* method.

**Remark 3.2.** *It is observed that the empirical Bayes method* (3.12) *may also induce sparse hyperparameter estimates here. This is because the log-determinant term* $\frac{1}{2}\log\det\Psi(\eta)$ *is known to promote low-rank structures of* $K(\eta)$ *which corresponds to a sparse* $\eta$ *estimate (Fazel et al., 2003). Nevertheless, the sparsity of the MAP estimator can be controlled by the rate parameter* $\lambda_\eta$*, recovering the empirical Bayes estimate by setting* $\lambda_\eta = 0$*.*

### 3.3.3 Numerical Results

This subsection compares the proposed multiple kernel design with sparse hyperparameter tuning to existing regularization formulations with the atomic structure. The least squares method without regularization and a single kernel design with the TC kernel are also compared as the baseline performance. Note that these two methods do not estimate a low-complexity model regarding the number of poles.

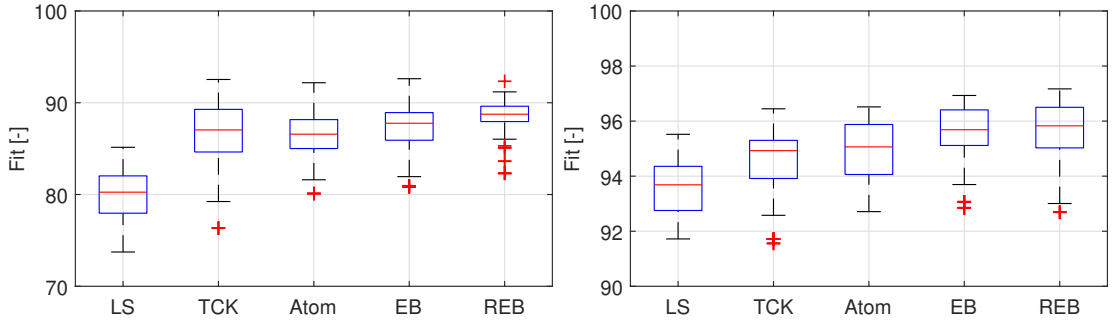
Specifically, the following five identification schemes are compared. The least squares method (*LS*) corresponds to the estimate $\hat{\mathbf{g}}^{\mathrm{LS}}$ in (3.3). The system is also identified with a TC kernel (*TCK*) regularization. The hyperparameters are selected by the empirical Bayes method with non-convex optimization. This is also the defaulted identification method used in the MATLAB command `impulseest`. The atomic norm method (*Atom*) applies the discretized atomic norm regularization in Shah et al. (2012). *Atom* uses a set of atomic transfer functions characterized by the poles $k = \alpha\exp(j\beta)$, where $\beta = [0 : \pi/15 : \pi]$ in the MATLAB notation. The magnitude $\alpha$ is in a 15-point logarithmic grid of base $10^6$ between 0.8 and 1 to obtain a denser grid near $\alpha = 1$. The empirical Bayes method (*EB*) uses the hyperparameter estimate (3.12) without explicitly exploiting the sparse kernel structure. The regularized empirical Bayes method (*REB*) refers to the method proposed in this section. Both *EB* and *REB* regularize the problem with the multiple kernel design (3.17) with the same set of poles as *Atom*. This gives a total of $n_\eta = 240$ kernels.

To highlight the characteristics of the bias-variance trade-off in these methods, Monte Carlo simulations are conducted on a benchmark system under i.i.d Gaussian noise of two different levels ($\sigma^2 = 0.01, 0.001$) with 150 different noise realizations each. The transfer function of the chosen fourth-order system is

$$G_2(q) = \frac{0.1159(q^3 + 0.5q^2)}{q^4 - 2.2q^3 + 2.42q^2 - 1.87q + 0.7225}, \tag{3.26}$$

which is one of the benchmark systems tested in Pillonetto and De Nicolao (2010). The system has been normalized to have an $\mathcal{H}_2$-norm of 1. The input to the system is Gaussian with $u_t^d \sim \mathcal{N}(0,1)$. The length of the identification data is $N = 150$, and the order of the FIR model is $n_g = 50$. For *EB* and *REB*, the noise variance $\sigma^2$ is estimated from the variance of the residuals in *LS*. For

*Atom* and *REB*, the weighting of the $l_1$-norm regularization $\lambda$ and the rate parameter $\lambda_\eta$ are cross-validated over a five-point grid of `logspace(0,4,5)` and `logspace(-1,1,5)` in the MATLAB notation, respectively. The DCP optimization problem in *REB* is solved by a fixed number of five iterations.




(a) $\sigma^2 = 0.01$ (b) $\sigma^2 = 0.001$

Figure 3.1: Comparison of fitting performance under different noise levels. The last three methods estimate a low-complexity model with atomic structure.

Table 3.1: Bias-variance trade-off of different estimates.

| | **LS** | **TCK** | **Atom** | **EB** | **REB** |
|---|---|---|---|---|---|
| $\sigma^2 = 0.01$ | | | | | |
| Bias$^2$ $[\times 10^{-4}]$ | 1.6 | 8.5 | 11.2 | 5.6 | 8.4 |
| Var $[\times 10^{-4}]$ | 61.0 | 23.6 | 17.1 | 20.8 | 12.3 |
| MSE $[\times 10^{-4}]$ | 62.6 | 32.1 | 28.3 | 26.4 | **20.8** |
| $\sigma^2 = 0.001$ | | | | | |
| Bias$^2$ $[\times 10^{-4}]$ | 0.15 | 0.96 | 1.24 | 0.37 | 0.68 |
| Var $[\times 10^{-4}]$ | 6.23 | 3.79 | 2.97 | 2.98 | 2.52 |
| MSE $[\times 10^{-4}]$ | 6.38 | 4.75 | 4.21 | 3.35 | **3.20** |

First, the fitting performance of the five methods is compared in Figure 3.1 using boxplots. As can be seen from Figure 3.1, *REB* achieves the best fitting performance at both noise levels. Compared to *REB*, the performance of *EB* is poor under the higher noise level, whereas *Atom* and *TCK* perform worse under the lower noise level. In both cases, *LS* fails to perform well without regularization.

Furthermore, the bias and the variance of the estimates are calculated in Table 3.1. It can be seen, as discussed in Pillonetto et al. (2014), that the bias-variance trade-off is controlled by the amount of regularization imposed. The MSE of *LS* is dominated by the variance since no regularization is imposed. Note that there is an inherent bias induced by the impulse response truncation. *Atom* induces the highest amount of bias with direct $l_1$-norm regularization. The proposed method *REB* imposes more regularization than *EB* with an additional sparsity regularization in

hyperparameter tuning, but less regularization compared to the direct sparsity regularization on the impulse response estimation as in *Atom*. This characteristic leads to an appropriate balance in the bias-variance trade-off as seen from the MSE values. This result agrees with the discussion in Remark 3.1 and Section 3.3.2.

## 3.4 Error Bounds with Unknown Hyperparameters

### 3.4.1 Pitfalls with Error Bounds from Posterior Covariances

The kernel-based method has shown remarkable performance in linear system identification, in terms of the nominal estimate (3.6). However, the stochastic error bound (3.10) is only rigorously valid when considering a random impulse response model subject to an exact prior distribution with exact hyperparameters. On the other hand, in practical system identification applications, a fixed plant is usually considered, and hyperparameters are estimated as the most probable ones if the impulse response is drawn from the prior distribution with the assumed structure. When the estimated hyperparameters $\hat{\eta}$ are used, directly using (3.10) to provide a stochastic model for a fixed plant can be problematic, as shown in the following example.

**Example 3.1.** *(Error bounds with estimated hyperparameters) Consider two second-order systems*

$$G_3(q) = \frac{0.4888}{q^2 - 1.8q + 0.9^2}, \quad G_4(q) = \frac{0.0616}{q^2 - q + 0.9^2}, \tag{3.27}$$

*with two different noise levels $\sigma^2 = 0.1$ and 0.5. Both systems have two poles of magnitude 0.9: $G_3(q)$ has two real poles at 0.9; $G_4(q)$ has a pair of complex poles with a real part of 0.5. The systems have been normalized to have an $\mathcal{H}_2$-norm of 1.*

*Stochastic models given by the error bound* (3.10) *with $\hat{\eta}$ are analyzed by 1000 Monte Carlo simulations with TC kernels. Different unit Gaussian inputs are used to generate the identification data in each run. Figure 3.2 shows the empirical probabilities of the error bounds containing the true impulse responses with $\delta = 0.1$ and identification parameters $N = 200$ and $n_g = 50$. Table 3.2 shows the empirical probabilities of violating the element-wise bounds.*

Table 3.2: Empirical probability of bound violations and standard deviations of hyperparameter estimation.

| $\delta = 0.1$ | % bound violations | STD($\hat{c}_K$) | STD($\hat{\lambda}_K$) |
|---|---|---|---|
| (a) $G_3, \sigma^2 = 0.1$ | 13.2% | 0.0052 | 0.0069 |
| (b) $G_4, \sigma^2 = 0.1$ | 29.8% | 0.2191 | 0.0204 |
| (c) $G_3, \sigma^2 = 0.5$ | 24.6% | 0.0010 | 0.0313 |
| (d) $G_4, \sigma^2 = 0.5$ | 60.1% | 0.0242 | 0.0373 |

It can be seen that except for the case of $G_3(q)$ with low noise, the magnitudes of the errors are significantly underestimated in the other three cases, with bound violation probabilities much

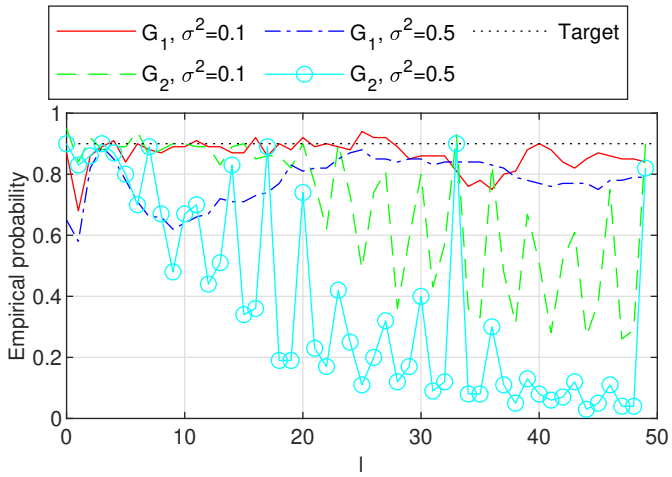



Figure 3.2: Empirical probability of error bounds containing the true parameters using estimated hyperparameters. $l$: index of the impulse response vector.

larger than the target value of $\delta = 0.1$. This indicates that the error bounds based on estimated hyperparameters are unreliable when the impulse response is lightly damped and/or the SNR is poor.

To investigate the reason why the error bounds are inaccurate under these cases, Table 3.2 also shows the standard deviations of the estimated hyperparameters, and Figure 3.3 plots the marginal probability density (3.13) with respect to the hyperparameters in one representative simulation. It can be seen that in cases (b), (c), and (d), where the error bounds based on estimated hyperparameters are inaccurate, the variances of the estimated hyperparameters are more significant than those in case (a), and the marginal probability density is not strongly localized. This suggests that the estimated hyperparameters can be inaccurate, leading to the error bounds' misspecification.

### 3.4.2 Worst-Case Posterior Variances

To solve the problem of quantifying error bounds with unknown hyperparameters, we first bound the true hyperparameters using the measured data. Consider the following assumption.

**Assumption 3.1.** *The kernel structure $K(\eta)$ is assumed to be correct with unknown true hyperparameters $\eta_0$. The hyperprior $p(\eta)$ is known.*

The hyperprior $p(\eta)$ can be selected as a uniform distribution if no additional knowledge about the hyperparameters is available. The distribution of hyperparameters conditioned on the measured data is given by

$$p(\eta|\mathbf{u}^d,\mathbf{y}^d) = \frac{p(\mathbf{y}^d|\mathbf{u}^d,\eta)p(\eta)}{\int_{\eta\in\mathcal{E}} p(\mathbf{y}^d|\mathbf{u}^d,\eta)p(\eta)\,\mathrm{d}\eta}, \tag{3.28}$$

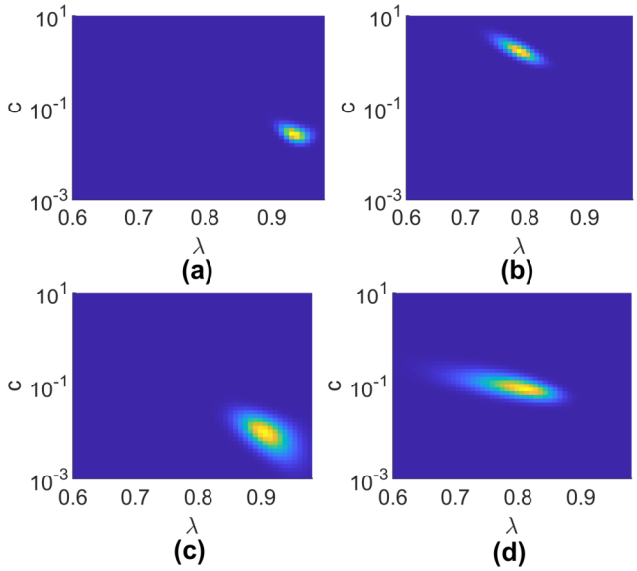



Figure 3.3: Marginal probability density with respect to hyperparameters. (a) $G_3, \sigma^2 = 0.1$, (b) $G_4, \sigma^2 = 0.1$, (c) $G_3, \sigma^2 = 0.5$, (d) $G_4, \sigma^2 = 0.5$. Yellow: higher value, blue: lower value.

where $p(\mathbf{y}^d|\mathbf{u}^d, \eta)$ is given in (3.13). This leads to

$$\mathbb{P}(\eta_0 \in [\eta_1, \eta_2]) = \frac{\int_{\eta \in [\eta_1, \eta_2]} p(\mathbf{y}^d|\mathbf{u}^d, \eta) p(\eta)\, \mathrm{d}\eta}{\int_{\eta \in \mathcal{E}} p(\mathbf{y}^d|\mathbf{u}^d, \eta) p(\eta)\, \mathrm{d}\eta} =: 1 - \delta', \tag{3.29}$$

where $\eta_i := [c_{K,i}\ \lambda_{K,i}]^\top$, $i = 1, 2$ and

$$[\eta_1, \eta_2] := \left\{ \eta = [c_K\ \lambda_K]^\top \,|\, c_{K,1} \le c_K \le c_{K,2}, \lambda_{K,1} \le \lambda_K \le \lambda_{K,2} \right\} \tag{3.30}$$

is a rectangular set. By choosing a small $\delta'$, (3.29) establishes a high-probability set for the true hyperparameters.

Then, we investigate the effect of hyperparameters on the stochastic model to find the worst-case posterior variances $\Sigma_{l,l}$ within the set $[\eta_1, \eta_2]$. A uniform bound is derived analytically using the following lemma for DI and TC kernels.

**Lemma 3.2.** *The matrix inequality* $\Sigma(\eta_1) \preccurlyeq \Sigma(\eta_2)$ *is satisfied when* $\left(\frac{\lambda_{K,2}}{\lambda_{K,1}}\right)^\gamma c_{K,1} \le c_{K,2}$, $\lambda_{K,1} \le \lambda_{K,2}$, *with* $\gamma = 0$ *for DI kernels and* $\gamma = -1/\ln \lambda_{K,2} - 1$ *for TC kernels.*

*Proof.* The result is trivial for DI kernels. For TC kernels, define $M(\mathbf{m}_n) \in \mathbb{R}^{n \times n}$, $\mathbf{m}_n := [m_1\ m_2\ \dots\ m_n]^\top$ with $M_{i,j}(\mathbf{m}_n) := m_{\max(i,j)}$. We first prove that

$$\det M(\mathbf{m}_n) = m_n \prod_{i=1}^{n-1} (m_i - m_{i+1}). \tag{3.31}$$

For $n = 1,2$, $\det M(\mathbf{m}_1) = m_1$, $\det M(\mathbf{m}_2) = m_2(m_1 - m_2)$ satisfy (3.31). Suppose (3.31) is satisfied for $n = l-1, l$. Define

$$\mathbf{m}_{l\setminus i} = [m_1 \ \dots \ m_{i-1} \ m_{i+1} \ \dots \ m_l]^\top. \tag{3.32}$$

From the definition of the determinant, we have $\det M(\mathbf{m}_l) = m_l \sum_{i=1}^{l} (-1)^{l-i} \det M(\mathbf{m}_{l\setminus i})$.

For $n = l+1$,

$$\begin{aligned}
\det M(\mathbf{m}_{l+1}) &= m_{l+1}\left(\det M(\mathbf{m}_l) + \sum_{i=1}^{l}(-1)^{l+1-i}\det M(\mathbf{m}_{l+1\setminus i})\right) \\
&= m_{l+1}\left(1 - \frac{m_{l+1}(m_l - m_{l+1})}{m_l} - \frac{m_{l+1}^2}{m_l^2}\right)\det M(\mathbf{m}_l) \\
&= m_{l+1}\prod_{i=1}^{l}(m_i - m_{i+1})
\end{aligned} \tag{3.33}$$

satisfies (3.31). This proves (3.31) by induction.

Using Sylvester's criterion, $M(\mathbf{m}_n)$ is positive semidefinite iff $\det M(\mathbf{m}_l) \geq 0, \forall\, l = 1,\dots,n$. This requires

$$m_i - m_{i+1} \geq 0, \quad \forall\, i = 1,\dots,n-1. \tag{3.34}$$

Define $\eta_2' := \left[\left(\frac{\lambda_{K,2}}{\lambda_{K,1}}\right)^\gamma c_{K,1} \ \lambda_{K,2}\right]^\top$. Since $\left(\frac{\lambda_{K,2}}{\lambda_{K,1}}\right)^\gamma c_{K,1} \leq c_{K,2}$, we have $K(\eta_2) \succcurlyeq K(\eta_2')$. Define $M(\mathbf{m}_{n_g}) := K(\eta_2') - K(\eta_1)$ by choosing $m_i = \left(\frac{\lambda_{K,2}}{\lambda_{K,1}}\right)^\gamma c_{K,1}\lambda_{K,2}^i - c_{K,1}\lambda_{K,1}^i$. So $K(\eta_2') - K(\eta_1) \succcurlyeq 0$ is equivalent to

$$\lambda_{K,2}^{1+\gamma} - \lambda_{K,1}^{1+\gamma} \geq \lambda_{K,2}^{2+\gamma} - \lambda_{K,1}^{2+\gamma} \geq \dots \geq \lambda_{K,2}^{n_g+\gamma} - \lambda_{K,1}^{n_g+\gamma}. \tag{3.35}$$

Note that $f(x) = \lambda_{K,2}^x - \lambda_{K,1}^x$ is monotonically non-increasing for $x \geq -1/\ln\lambda_{K,2}$, $\forall\, \lambda_{K,2} \geq \lambda_{K,1}$. This indicates that (3.35) is satisfied for $\gamma \geq -1/\ln\lambda_{K,2} - 1$. Therefore, $K(\eta_2) \succcurlyeq K(\eta_2') \succcurlyeq K(\eta_1)$ for $\gamma = -1/\ln\lambda_{K,2} - 1$, which leads to

$$\left(\Phi^\top\Phi + \sigma^2 K^{-1}(\eta_2)\right)^{-1} \succcurlyeq \left(\Phi^\top\Phi + \sigma^2 K^{-1}(\eta_1)\right)^{-1}. \tag{3.36}$$

This directly proves the lemma. □

From Lemma 3.2, we have

$$\Sigma(\eta_0) \overset{1-\delta'}{\preccurlyeq} \sigma^2\left(\Phi^\top\Phi + \sigma^2\left(\frac{\lambda_{K,1}}{\lambda_{K,2}}\right)^\gamma K^{-1}(\eta_2)\right)^{-1} =: \bar{\Sigma}. \tag{3.37}$$

So, the posterior variances with true hyperparameters $\eta_0$ can be uniformly bounded by

$$\Sigma_{l,l}(\eta_0) \overset{1-\delta'}{\leq} \bar{\Sigma}_{l,l} =: \sigma_l^2. \tag{3.38}$$

For a general kernel structure, the bound can be computed element-wise by directly solving the optimization problem:

$$\sigma_l^2 = \max_{\eta \in [\eta_1, \eta_2]} \Sigma_{l,l}(\eta). \tag{3.39}$$

### 3.4.3 Stochastic Error Bounds

We are now ready to present the main result of this section.

**Theorem 3.1.** *Under Assumption 3.1, the impulse response estimate* (3.6) *with estimated hyperparameters* $\hat{\eta}$ *admits the following stochastic element-wise error bound:*

$$\mathbb{P}\left(|\hat{g}_l(\hat{\eta}) - g_l| \leq \bar{\mu}\sigma_l\right) \geq (1-\delta)(1-\delta'), \tag{3.40}$$

*where* $\bar{\mu} := \mu_\delta + \frac{2}{\sigma}\left\|\mathbf{y}^d\right\|_S$ *and* $S := \Phi\left(\Phi^\top\Phi\right)^{-1}\Phi^\top$*, if* $\hat{\eta} \in [\eta_1, \eta_2]$.

*Proof.* The estimation error is decomposed as

$$\begin{aligned} |\hat{g}_l(\hat{\eta}) - g_l| \;&\leq\; |\hat{g}_l(\hat{\eta}) - \hat{g}_l(\eta_0)| + |\hat{g}_l(\eta_0) - g_l| && (3.41)\\ &\overset{1-\delta}{\leq}\; |\hat{g}_l(\hat{\eta}) - \hat{g}_l(\eta_0)| + \mu_\delta\sqrt{\Sigma_{l,l}(\eta_0)}, && (3.42) \end{aligned}$$

where the two terms are due to misspecified hyperparameters and measurement noise, respectively.

Define the posterior kernel

$$k_\eta^p(x,x') := k_\eta(x,x') - \mathbf{k}_x(\eta)\left(K(\eta) + \sigma^2\left(\Phi^\top\Phi\right)^{-1}\right)^{-1}\mathbf{k}_x(\eta)^\top. \tag{3.43}$$

Note that $k_\eta^p(i,j) = \Sigma_{i,j}(\eta)$. The associated RKHS is denoted as $\mathcal{H}_\eta^p$. It is easy to see that $g_\eta^\star(\cdot) \in \mathcal{H}_\eta^p$ and $\left\|g_\eta^\star(\cdot)\right\|_{\mathcal{H}_\eta^p}^2 = \hat{\mathbf{g}}^\top(\eta)\Sigma^{-1}(\eta)\hat{\mathbf{g}}(\eta)$. Note the reproducing property of the RKHS $g_\eta^\star(x) = \langle g_\eta^\star(\cdot), k_\eta^p(\cdot,x)\rangle_{\mathcal{H}_\eta^p}$, where $\langle\cdot,\cdot\rangle_{\mathcal{H}_\eta^p}$ denotes the inner product in $\mathcal{H}_\eta^p$. From the Cauchy–Schwarz inequality, we have $\left|g_\eta^\star(x)\right| \leq k_\eta^p(x,x)^{\frac{1}{2}}\left\|g_\eta^\star(\cdot)\right\|_{\mathcal{H}_\eta^p}$. This leads to

$$\begin{aligned} |\hat{g}_l(\eta)|^2 &\leq \Sigma_{l,l}(\eta)\hat{\mathbf{g}}^\top(\eta)\Sigma^{-1}(\eta)\hat{\mathbf{g}}(\eta)\\ &= \frac{1}{\sigma^2}\Sigma_{l,l}(\eta)\left(\mathbf{y}^d\right)^\top\Phi\left(\Phi^\top\Phi + \sigma^2K^{-1}(\eta)\right)^{-1}\Phi^\top\mathbf{y}^d \qquad (3.44)\\ &\leq \Sigma_{l,l}(\eta)\left\|\mathbf{y}^d\right\|_S^2/\sigma^2. \end{aligned}$$

Since $\hat{\eta} \in [\eta_1, \eta_2]$, we have $\Sigma_{l,l}(\hat{\eta}) \leq \sigma_l^2$. This leads to $|\hat{g}_l(\hat{\eta})|^2 \leq \frac{\sigma_l^2}{\sigma^2}\left\|\mathbf{y}^d\right\|_S^2$ and $|\hat{g}_l(\eta_0)|^2 \overset{1-\delta'}{\leq} \frac{\sigma_l^2}{\sigma^2}\left\|\mathbf{y}^d\right\|_S^2$. Then,

$$|\hat{g}_l(\hat{\eta}) - \hat{g}_l(\eta_0)| \leq |\hat{g}_l(\hat{\eta})| + |\hat{g}_l(\eta_0)| \overset{1-\delta'}{\leq} \frac{2\sigma_l}{\sigma}\left\|\mathbf{y}^d\right\|_S. \tag{3.45}$$

From (3.29), (3.38), (3.39), we have $\mu_\delta \sqrt{\Sigma_{l,l}(\eta_0)} \overset{1-\delta'}{\leq} \mu_\delta \sigma_l$. This, together with (3.42) and (3.45), proves Theorem 3.1. □

Theorem 3.1 provides high-probability error bounds with unknown hyperparameters by replacing the estimated posterior variance $\Sigma_{l,l}$ with its worst-case counterpart $\sigma_l^2$. We note the following remarks on Theorem 3.1.

**Remark 3.3.** *For DI and TC kernels, by modifying the last inequality in* (3.44)*, the bound in Theorem 3.1 can be tightened by choosing* $S = \Phi \left( \Phi^\top \Phi + \sigma^2 \left( \frac{\lambda_{K,1}}{\lambda_{K,2}} \right)^\gamma K^{-1}(\eta_2) \right)^{-1} \Phi^\top$.

**Remark 3.4.** *Theorem 3.1 still holds when more hyperparameters are involved with minor modifications to Lemma 3.2 if needed. So, the proposed approach can be extended to consider unknown noise levels and ARX models with an additional kernel on the autoregressive output terms.*

**Remark 3.5.** *Although Theorem 3.1 improves the bounds in Capone et al. (2022), the constant* $\bar{\mu}$ *is still quite conservative, mainly due to the triangle equality in* (3.45)*. Such conservativeness is often observed in GP error bounds, so a much smaller scaling factor is often selected in practical applications (Capone et al., 2022; Berkenkamp et al., 2017; Umlauft et al., 2017), despite that this invalidates the theoretical guarantees. As will be seen in Section 3.4.5,* $\bar{\mu} = \mu_\delta$ *is used in numerical simulation.*

### 3.4.4 Selecting the Set of Hyperparameters

Theorem 1 holds for any choices of $\eta_1, \eta_2$ that satisfy (3.29) and $\hat{\eta} \in [\eta_1, \eta_2]$. To obtain the tightest bound, $\eta_1, \eta_2$ can be selected by optimization. For DI and TC kernels, the total magnitude of the bounds $\sum_{l=0}^{n_g-1} \bar{\mu}\sigma_l$ can be minimized. From (3.37) and (3.38), this is equivalent to solving

$$\min_{\eta_1,\eta_2} \quad \left( \frac{\lambda_{K,2}}{\lambda_{K,1}} \right)^\gamma \operatorname{tr}(K(\eta_2)) \tag{3.46a}$$

$$\text{s.t.} \quad \frac{\int_{\eta \in [\eta_1,\eta_2]} p(\mathbf{y}^d|\mathbf{u}^d,\eta) p(\eta)\, \mathrm{d}\eta}{\int_{\eta \in \mathcal{E}} p(\mathbf{y}^d|\mathbf{u}^d,\eta) p(\eta)\, \mathrm{d}\eta} \geq 1-\delta',\ \hat{\eta} \in [\eta_1,\eta_2]. \tag{3.46b}$$

For a general kernel structure with element-wise bound (3.39), $\eta_1, \eta_2$ can be selected individually for each $l$ by solving the minimax problem:

$$\sigma_l^2 = \min_{\eta_1,\eta_2} \max_{\eta \in [\eta_1,\eta_2]} \Sigma_{l,l}(\eta) \quad \text{s.t. (3.46b).} \tag{3.47}$$

The algorithm to obtain the error bounds with unknown hyperparameters is summarized in Algorithm 3.1.

**Algorithm 3.1** Stochastic error bounds with unknown hyperparameters

1: Estimate $\hat{\eta}$ and obtain $\hat{\mathbf{g}}(\hat{\eta})$ from (3.6).
2: Calculate $\eta_1, \eta_2$ by solving (3.46) or (3.47).
3: Calculate $\sigma_l$, $l = 0, \ldots, n_g - 1$ from (3.38) or (3.39).
4: Obtain the element-wise error bounds from (3.40).

### 3.4.5 Numerical Results

The proposed bound is applied numerically by considering the same problem as in Example 3.1. Again, the practical scenario with fixed impulse responses is considered. The error bound (3.10) with estimated hyperparameters analyzed in Section 3.4.1 is termed the *vanilla kernel bound*, whereas the proposed bound in Section 3.4.3 is called the *robust kernel bound*. The *least-squares bound* (3.4) is also compared.

For computational efficiency, the optimization problems to find $\eta_1, \eta_2$ are solved by discretizing $\eta$. The nominal estimate and the estimated hyperparameters are obtained by `impulseest` in MATLAB. The inner problem in (3.47) is solved by `fmincon` in MATLAB. For the robust kernel bound, we select $\delta' = 0.1$ and $\bar{\mu} = \mu_\delta$.

Figure 3.4 compares the performance of different error bounds with a TC kernel design. For each case, the left figure shows representative identification results in one simulation, whereas the right figure shows the empirical probability of error bounds containing the true parameters from 1000 Monte Carlo simulations. The results show that the proposed robust kernel bounds are more conservative than the vanilla ones, especially under high noise. Still, they are much more reliable, with much higher empirical probabilities of containing the true parameters. On the other hand, the robust kernel bounds are still much tighter than the least-squares bounds.

Figure 3.5 shows the empirical probability with a SS kernel design. The robust kernel bounds are derived by selecting $\sigma_l$ from (3.47). Similar results to the TC kernel case are obtained, where the robust kernel bounds are much more reliable than the vanilla kernel bounds.

## 3.5 Summary

This chapter investigates two problems in the kernel-based identification of linear systems. The first part demonstrates that a low-order model can be obtained in kernel-based identification with appropriate multiple kernel design. Using optimal kernel design for first-order systems, low-complexity models correspond to sparse hyperparameter selection, which is imposed by a sparse hyperprior. Compared to direct $l_1$-norm regularization, this method achieves a more favorable bias-variance trade-off.

In the second part, a practical approach is provided to obtaining a reliable stochastic model centered around the nominal estimate of kernel-based system identification. Instead of constructing

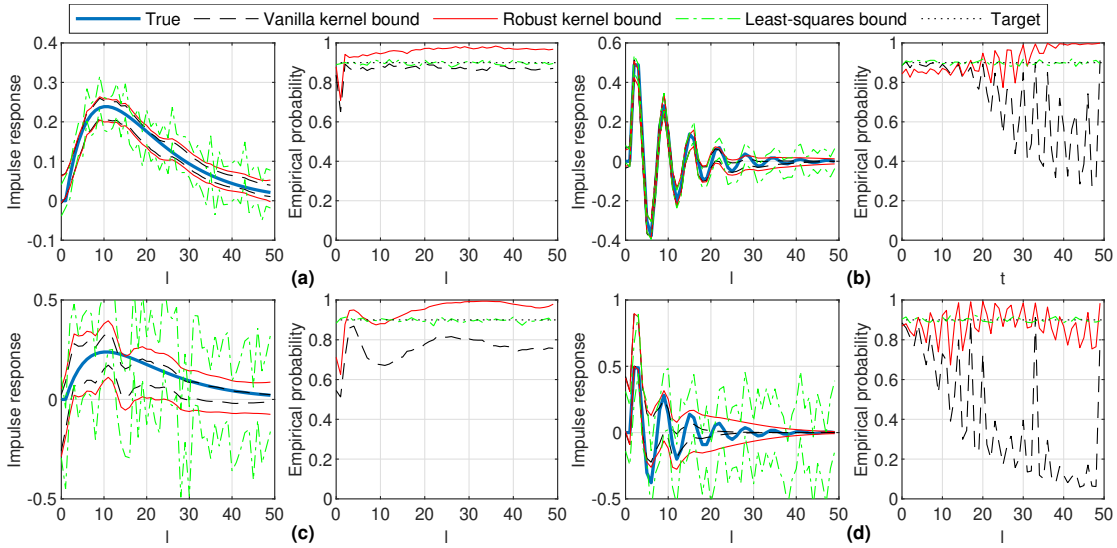



Figure 3.4: Comparison of different error bounds with TC kernels. (a) $G_3, \sigma^2 = 0.1$, (b) $G_4, \sigma^2 = 0.1$, (c) $G_3, \sigma^2 = 0.5$, (d) $G_4, \sigma^2 = 0.5$. Left: representative element-wise error bounds, right: the empirical probability of error bounds containing the true parameters. $l$: index of the impulse response vector.

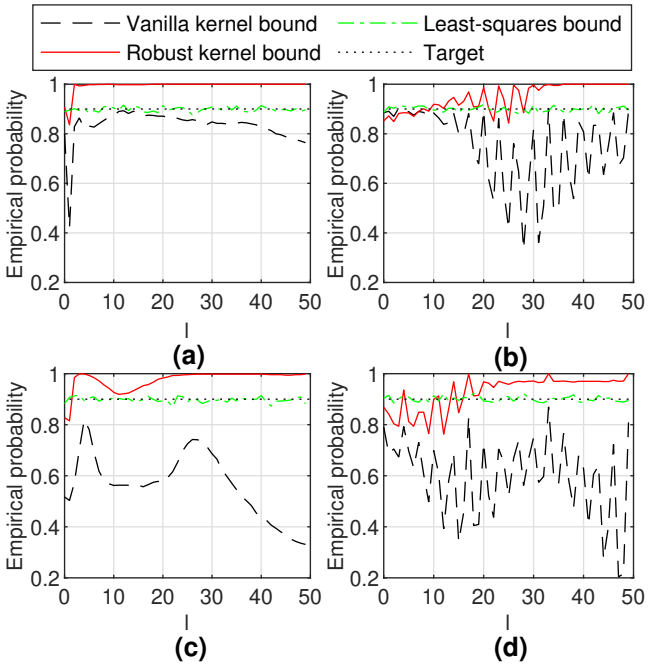



Figure 3.5: Empirical probability of error bounds containing the true parameters with SS kernels. (a) $G_3, \sigma^2 = 0.1$, (b) $G_4, \sigma^2 = 0.1$, (c) $G_3, \sigma^2 = 0.5$, (d) $G_4, \sigma^2 = 0.5$. $l$: index of the impulse response vector.

error bounds with estimated hyperparameters, which are too optimistic, the error bounds can be obtained from the worst-case posterior variances within a high-probability set of the true hyperparameters.

# Part II
# Nonparametric Prediction and Data-Driven Predictive Control

# 4 Nonparametric Trajectory Prediction with Stochastic Data

As discussed in Section 1.3, this chapter investigates the problem of directly predicting system responses from collected trajectory data. Based on the so-called Willems' fundamental lemma (WFL) (Willems et al., 2005; Markovsky and Dörfler, 2023; van Waarde et al., 2020), this problem has been well-studied for linear systems (Markovsky et al., 2005b; Markovsky and Rapisarda, 2008) when deterministic data are available with equivalence to the model-based predictors under mild conditions on the data quality (van Waarde et al., 2020). These results are summarized in Section 4.1.1.

However, one critical problem with the WFL is that the corresponding predictor becomes ill-defined when any uncertainty is present in the collected data. Optimal control design can still be conducted despite the ill-definedness by using the regularization technique to design the control cost under the robust control framework (Berberich et al., 2021; Huang et al., 2023; Coulson et al., 2022). Such methods are known as *direct* data-driven predictive control (DDPC) or data-enabled predictive control (DeePC) (Coulson et al., 2019). In this thesis, however, we focus on explicitly obtaining a well-defined nonparametric predictor in the presence of stochastic uncertainty, which is known as *indirect* DDPC. As will be detailed in Section 4.1.2, broadly speaking, algorithms to design such predictors can be divided into two categories, namely 1) by denoising the data and 2) by regularizing the predictor.

For the first category, the critical condition to guarantee the well-definedness of the predictor without regularization is the existence of a low-rank Hankel-structured data matrix. In other words, the data denoising problem can be equivalently posed as a low-rank Hankel matrix denoising problem. Although the optimal low-rank approximation of an unstructured matrix is well-known via singular value decomposition (SVD), the low-rank Hankel matrix denoising problem faces two additional issues. First, in matrix denoising, one is interested in minimizing the MSE of the estimate with respect to the noise-free matrix. However, the low-rank approximation does not guarantee any optimality on this since it also approximates the noise matrix. Remedies to this problem are investigated in Gavish and Donoho (2014, 2017); Nadakuditi (2014); Josse and Sardy (2015). Second, the optimal approximation does not preserve the Hankel structure.

Methods to preserve structural constraints in matrix approximation are summarized in Markovsky and Rapisarda (2008). Section 4.2 investigates the low-rank Hankel matrix denoising problem by combining approaches concerning both issues and comparing the performances of different algorithms numerically.

For the second category, the subspace predictor (Favoreel et al., 1999; Sedghizadeh and Beheshti, 2018) is commonly used in the literature, which can be interpreted as estimating a multi-step-ahead ARX model with close relations to the subspace identification algorithm (Fiedler and Lucia, 2021; Dörfler et al., 2023; Breschi et al., 2023). However, this predictor is not optimal for finite data length under realistic uncertainty assumptions. Instead, Section 4.3 proposes a maximum likelihood estimation (MLE) framework to obtain the regularized predictor under general uncertainty assumptions. The proposed predictor is named the signal matrix model (SMM). The SMM is shown to obtain more accurate output predictions than the subspace predictor. This predictor can be used to estimate the impulse response in system identification by conducting prediction with an impulse. This impulse response estimation approach guarantees an unbiased estimate without truncation errors or the requirement of knowing the input history. It demonstrates performance improvements over the least-squares estimate numerically. Since its proposal, SMM has been applied to other works as a nonparametric trajectory predictor, such as Furieri et al. (2023); Kergus and Gosea (2022). Input design for SMM is discussed in Iannelli et al. (2021a,b).

Finally, the prediction error of a general class of regularized predictors is quantified in Section 4.4. This provides confidence regions of output predictions, which is helpful to enforce safety-critical constraints in control design, as will be detailed in Chapter 5. The validity of the derived confidence regions is verified by numerical examples. In addition, this statistical framework allows approximate computation of the MSE of the predictor. In this way, another stochastic data-driven predictor can be designed to be optimal for minimizing the MSE. It is shown numerically that this minimum MSE predictor obtains marginally smaller prediction errors than the other stochastic predictors under high SNRs.

## 4.1 Willems' Fundamental Lemma and Data-Driven Prediction

### 4.1.1 Deterministic Data-Driven Prediction

Built originally on the notion of the persistency of excitation, the Willems' fundamental lemma (WFL) shows that all the behaviors of a linear system can be captured by a single sufficiently informative trajectory of the system when no uncertainty is present. This lemma was initially proposed in the context of behavioral system theory (Willems et al., 2005; Willems and Polderman, 1997), where systems are characterized by the subspace that contains all possible trajectories, with a more general version presented recently (Markovsky and Dörfler, 2021). It was later rephrased in the state-space context (De Persis and Tesi, 2020) and extended to multiple datasets (van Waarde et al., 2020).

In detail, we truncate the data sequence $(\mathbf{u}^d, \mathbf{y}^d)$ into length-$L$ sections:

$$\mathbf{z}_i^d := \operatorname{col}\left(u_i^d, \ldots, u_{i+L-1}^d, y_i^d, \ldots, y_{i+L-1}^d\right) \in \mathbb{R}^{(n_u+n_y)L}, \tag{4.1}$$

where $i = 1, \ldots, N-L+1$. Define the following mosaic Hankel matrix by concatenating the trajectory sections column-wise:

$$Z := \begin{bmatrix} \mathbf{z}_1^d & \mathbf{z}_2^d & \cdots & \mathbf{z}_M^d \end{bmatrix} := \left[\begin{array}{cccc} u_1^d & u_2^d & \cdots & u_M^d \\ \vdots & \vdots & \ddots & \vdots \\ u_L^d & u_{L+1}^d & \cdots & u_N^d \\ \hline y_1^d & y_2^d & \cdots & y_M^d \\ \vdots & \vdots & \ddots & \vdots \\ y_L^d & y_{L+1}^d & \cdots & y_N^d \end{array}\right] =: \begin{bmatrix} U \\ Y \end{bmatrix} \in \mathbb{R}^{(n_u+n_y)L \times M}, \tag{4.2}$$

where $M := N-L+1$. The data matrix $Z$ is termed the signal matrix in what follows.

Due to linearity, any linear combination of the length-$L$ trajectories $\mathbf{z}_i^d$ is still a possible trajectory of the system. On the other hand, all possible length-$L$ trajectories of the LTI system (1.2) formulate a subspace of $(n_u L + n_x)$ dimensions since all $n_u L$ inputs can be freely selected as well as the initial state of $n_x$ dimensions. Therefore, if the collected length-$L$ trajectories cover all $(n_u L + n_x)$ dimensions, i.e., $\operatorname{rank}(Z) = n_u L + n_x$, their linear combinations cover all possible trajectories. Then, by selecting $L = L_0 + L'$, the input-output mapping (1.9) can be characterized implicitly by the condition $\operatorname{col}(\mathbf{u}_{\text{ini}}, \mathbf{u}, \mathbf{y}_{\text{ini}}, \mathbf{y}) \in \operatorname{range}(Z)$.

The above idea is presented formally in the following theorem, based on the WFL for finite-dimensional LTI systems. These results hold exactly only when the system is noise-free, i.e., $\forall t, v_t = 0$.

**Theorem 4.1.** *Consider the finite-dimensional LTI system $(A,B,C,D)$ in* (1.2)*. Let $\left(u_i^d, x_i^d, y_i^d\right)_{i=1}^N$ be a noise-free input-state-output trajectory of the system. The following holds iff the signal matrix $Z$ defined in* (4.2) *satisfies*

$$\operatorname{rank}(Z) = n_u L + n_x. \tag{4.3}$$

1. *The matrix $\operatorname{col}(X_{\text{ini}}, U)$ has full row rank, where $X_{\text{ini}} := \begin{bmatrix} x_1^d & x_2^d & \cdots & x_M^d \end{bmatrix}$ collects the initial states of the trajectory sections (Corollary 2 in Willems et al. (2005), Theorem 1(i) in van Waarde et al. (2020), Lemma 1 in De Persis and Tesi (2020)).*
2. *The pair $(u_i, y_i)_{i=0}^{L-1}$ is an input-output trajectory of the system iff there exists $g \in \mathbb{R}^M$, such that $\operatorname{col}(u_0, \ldots, u_{L-1}, y_0, \ldots, y_{L-1}) = Zg$ (Theorem 1 in Willems et al. (2005), Theorem 1(ii) in van Waarde et al. (2020), Lemma 2 in De Persis and Tesi (2020)).*
3. *The vector $\mathbf{y} := \operatorname{col}(y_0, \ldots, y_{L'-1})$ is the unique output trajectory of the system with immediate past input-output trajectory $\mathbf{u}_{\text{ini}} := \operatorname{col}(u_{-L_0}, \ldots, u_{-1})$, $\mathbf{y}_{\text{ini}} := \operatorname{col}(y_{-L_0}, \ldots, y_{-1})$ and given input trajectory $\mathbf{u} := \operatorname{col}(u_0, \ldots, u_{L'-1})$, where $L_0$ is no smaller than the observability*

*index $l$ of the system and $L = L_0 + L'$, iff there exists $g \in \mathbb{R}^M$, such that*

$$\operatorname{col}(\mathbf{u}_{\text{ini}}, \mathbf{u}, \mathbf{y}_{\text{ini}}, \mathbf{y}) = Zg \tag{4.4}$$

*(Proposition 1 in Markovsky and Rapisarda (2008)).*

**Remark 4.1.** *The signal matrix $Z$ can be alternatively constructed as*

$$Z := \begin{bmatrix} \mathbf{z}_1^d & \mathbf{z}_{L+1}^d \cdots & \mathbf{z}_{(M-1)L+1}^d \end{bmatrix}, \tag{4.5}$$

*where $M := \lfloor N/L \rfloor$, which is known as the Page construction (Damen et al., 1982). As will be seen in Remark 4.5, this construction leads to simpler noise statistics with no repeated elements at the expense of poor data efficiency. It is useful in, for example, input design (Iannelli et al., 2021a). The length-$L$ trajectories $\mathbf{z}_i^d$ can also come from independent experiments. It was shown in van Waarde et al. (2020) that similar results to Theorem 4.1 still hold for Page matrices and/or multiple experiments.*

In the original work (Willems et al., 2005), instead of the rank condition (4.3), more conservative conditions are used, namely, the matrix pair $(A, B)$ is controllable and the input is persistently exciting of order $(L + n_x)$. These constraints are relaxed in Theorem 1 of Yu et al. (2021) before the necessary and sufficient rank condition is finally given in Corollary 19 of Markovsky and Dörfler (2023). A necessary condition for the rank condition to hold is $M = N - L + 1 \geq n_u L + n_x$, which offers a lower bound on the trajectory length. Despite its conservativeness, the persistency of excitation test is still useful when the data are contaminated by uncertainties and the signal matrix loses the low-rank structure, as will be seen in Section 4.1.2. So, we still define persistency of excitation as follows.

**Definition 4.1.** *A signal trajectory $(x_i)_{i=1}^N \in \mathbb{R}^n \times \{1, \ldots, N\}$ is said to be persistently exciting of order $L$ if*

$$X := \begin{bmatrix} x_1 & x_2 & \cdots & x_M \\ \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{bmatrix} \in \mathbb{R}^{nL \times M} \tag{4.6}$$

*has full row rank (Willems et al., 2005).*

In Theorem 4.1, parts 1 and 2 state the original Willems' fundamental lemma (WFL), whereas part 3 directly motivates the design of a deterministic data-driven predictor. Define a partition of $Z$ as $Z =: \operatorname{col}(U_p, U_f, Y_p, Y_f)$, where $U_p \in \mathbb{R}^{n_u L_0 \times M}$, $U_f \in \mathbb{R}^{n_u L' \times M}$, $Y_p \in \mathbb{R}^{n_y L_0 \times M}$, $Y_f \in \mathbb{R}^{n_y L' \times M}$. The deterministic data-driven predictor can be constructed by a two-step approach with $g$ as the intermediate parameter, as shown in Algorithm 4.1. Although any solution to (4.9) is applicable (Proposition 1 in Markovsky and Rapisarda (2008)), the pseudoinverse solution is the most commonly used, which leads to the following input-output mapping:

$$\mathbf{y} = \mathcal{F}_Z(\mathbf{u}; \mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}) = Y_f g_{\text{pinv}}, \tag{4.7}$$

where

$$g_{\text{pinv}} := \text{col}\left(U_p, U_f, Y_p\right)^{\dagger} \text{col}\left(\mathbf{u}_{\text{ini}}, \mathbf{u}, \mathbf{y}_{\text{ini}}\right). \tag{4.8}$$

**Algorithm 4.1** Deterministic data-driven predictor (Markovsky et al., 2005b)

1: **Given:** signal matrix $Z$
2: **Input:** $\mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}, \mathbf{u}$
3: Solve the linear system

$$\text{col}\left(U_p, U_f, Y_p\right) g = \text{col}\left(\mathbf{u}_{\text{ini}}, \mathbf{u}, \mathbf{y}_{\text{ini}}\right) \tag{4.9}$$

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

### 4.1.2 Towards Stochastic Data-Driven Trajectory Prediction

A strong assumption in Theorem 4.1 is that the data are deterministic without uncertainties, under which the signal matrix is rank deficient with (4.3). This guarantees that, though (4.9) is a highly underdetermined linear system when a large dataset is available, the input-output mapping from $\mathbf{u}$ to $\mathbf{y}$ is still well-defined, i.e., the prediction $\mathbf{y}$ is unique for any solution to (4.9). However, when the data are contaminated, the signal matrix $Z$ has full row rank almost surely, and thus for any $\mathbf{u}$ and $\mathbf{y}$, there exists $g$ that satisfies (4.4) almost surely. This means that the output prediction from Algorithm 4.1 can be any trajectory by manipulating the solution of $g$, so the input-output mapping becomes ill-defined. It is also not clear if satisfying condition (4.9) is still necessary with stochastic data since Theorem 4.1.3 does not hold exactly.

To recover the well-definedness of the predictor in the presence of stochastic uncertainties, two types of modifications to Algorithm 4.1 are investigated.

**Structured low-rank matrix denoising.** This method formulates a structured low-rank matrix denoising problem to recover the rank condition (4.3) and thus the well-definedness of the predictor.

The general structured low-rank matrix denoising problem is formulated as follows. For a set of structured matrix $\mathbb{M}^{m\times n} \subseteq \mathbb{R}^{m\times n}$, consider the problem of estimating an unknown matrix $X \in \mathbb{M}^{m\times n}$ from a noisy measurement $\tilde{X} = X + \sigma V$, where $\sigma$ is the noise level and $V \in \mathbb{M}^{m\times n}$ is a stochastic noise matrix with zero mean. It is known that the unknown matrix $X$ has the low-rank property $\text{rank}(X\Pi) = r$, where $\Pi \in \mathbb{R}^{n\times n}$ is a known transformation matrix. Without loss of generality, let $r < m \leq n$ and $\beta := m/n \in (0,1]$. To obtain the optimal estimate of the noise-free matrix $X$, we are interested in minimizing the MSE of the estimate:

$$\text{MSE}\left(\hat{X}\right) := \mathbb{E}\left(\left\|X - \hat{X}\right\|_F^2\right), \tag{4.10}$$

where the estimate $\hat{X}$ is a function of the measurement $\tilde{X}$. This problem will be referred to as the

denoising problem.

In this thesis, we are restricted to denoising output trajectories of SISO systems and consider the case where $\mathbb{M}^{m\times n}$ is the set of $m$-by-$n$ Hankel matrices. In particular, let $Y^0$ be constructed similar to $Y$ but with noise-free outputs $(y_i^{d,0})_{i=1}^N$ instead of contaminated outputs $\left(y_i^d\right)_{i=1}^N$. Since $\operatorname{rank}(\operatorname{col}\left(U,Y^0\right)) = n_u L + n_x$, the projection of $Y^0$ onto the null space of $U$ has a low rank, i.e., the low-rank Hankel matrix denoising problem can be formulated with

$$X = Y^0,\ \tilde{X} = Y,\ \Pi = \Pi_U^{\perp},\ r = n_x, \tag{4.11}$$

where $\Pi_U^{\perp}$ spans the null space of $U$ and can be calculated as $\Pi_U^{\perp} = \mathbb{I} - U^\top (UU^\top)^{-1} U$.

**Remark 4.2.** *One special case of the output denoising problem is impulse response denoising. Consider the case where the output trajectory to be denoised is the first-N impulse response coefficients $(g_i)_{i=0}^{N-1}$ of the system, measured with additive noise: $\hat{g}_i = g_i + v_i$. Similar to $Y^0$ and $Y$, construct Hankel matrices with $(g_i)_{i=0}^{N-1}$ and $(\hat{g}_i)_{i=0}^{N-1}$ and denote them by $H_g$ and $H_{\hat{g}}$, respectively. The matrix $H_g$ is rank-deficient with* $\operatorname{rank}(H_g) = n_x$ *(Fazel et al., 2003). This leads to the denoising problem with*

$$X = H_g,\ \tilde{X} = H_{\hat{g}},\ \Pi = \mathbb{I},\ r = n_x. \tag{4.12}$$

*This special case has particular applications in frequency-domain subspace identification (McKelvey et al., 1996) and model order reduction (Markovsky et al., 2005a).*

This problem is also studied in the intersection algorithm of subspace identification (Moonen et al., 1989), where a low-order subspace of the signal matrix corresponding to a low state dimension is sought. Although a state-space realization is not required for data-driven prediction, prediction results with the denoised signal matrix are equivalent to those of the model-based predictor with state-space models identified by subspace identification (Fiedler and Lucia, 2021).

Details about solving the denoising problem are investigated in Section 4.2.

**Indirect data-driven prediction.** This method aims to obtain a unique predictor by finding a unique solution to the intermediate vector $g$. This turns out to be a difficult problem. The output noise leads to uncertainties in both the output signal matrix $Y$ and the output initial condition $\mathbf{y}_{\text{ini}}$. So the input equation $\operatorname{col}(U_p, U_f)\, g = \operatorname{col}(\mathbf{u}_{\text{ini}}, \mathbf{u})$ in (4.9) still holds exactly, but the past output equation $Y_p g = \mathbf{y}_{\text{ini}}$ includes noise on both sides. So, if we pose the problem as a parameter estimation problem of $g$, it becomes an error-in-variables problem. To make matters worse, there does not exist a unique true parameter, but a subspace of true parameters satisfying (4.9) in the noise-free case, and the prediction accuracy is evaluated on a projection ($Y_f$) of $g$ which is also unknown.

Despite the difficulty, this problem is often approached by solving optimization problems that find the optimal $g$ with respect to some statistical or empirical objectives. This type of predictors

usually shares the following form:

$$\hat{\mathbf{y}} = \mathcal{F}_Z(\cdot) = Y_f g, \tag{4.13a}$$

$$\operatorname{col}(U_p, U_f, Y_p)\, g = \operatorname{col}(\mathbf{u}_{\text{ini}}, \mathbf{u}, \mathbf{y}_{\text{ini}} + \delta). \tag{4.13b}$$

Here, the notation $\hat{\mathbf{y}}$ is used to indicate that the stochastic prediction is only an estimate of the true output trajectory $\mathbf{y}$. The slack variable $\delta$ is introduced to compensate for the error in both $Y_p$ and $\mathbf{y}_{\text{ini}}$. The predictors then propose different strategies for balancing the magnitude of $g$ and the slack variable $\delta$, a particular form of which is

$$\mathcal{F}_Z(\cdot) = Y_f \operatorname*{argmin}_g \ \|\delta\|_S^2 + \lambda \|g\|_2^2 \qquad \text{s.t.} \quad (4.13b), \tag{4.14}$$

where $S \in \mathbb{S}_{++}^{n_y L_0}$ and $\lambda \in \mathbb{R}_{++}$ are the design parameters. With an abuse of notation, $\operatorname{argmin}_g$ denotes the optimal solution of $g$ for the program depending on both $g$ and $\delta$. Note that the optimization problem (4.14) is a strongly convex quadratic programming (QP) problem with only equality constraints, the optimality conditions of which are:

$$\begin{bmatrix} F & U^\mathsf{T} \\ U & \mathbb{0} \end{bmatrix} \begin{bmatrix} g \\ \nu \end{bmatrix} = \begin{bmatrix} Y_p^\mathsf{T} S \mathbf{y}_{\text{ini}} \\ \tilde{\mathbf{u}} \end{bmatrix}, \tag{4.15}$$

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

$$g^\star = R_1 \tilde{\mathbf{u}} + R_2 \mathbf{y}_{\text{ini}}, \tag{4.16}$$

where

$$R_1 := F^{-1} U^\top (U F^{-1} U^\top)^{-1}, \tag{4.17}$$

$$R_2 := \left( F^{-1} - F^{-1} U^\top (U F^{-1} U^\top)^{-1} U F^{-1} \right) Y_p^\top S. \tag{4.18}$$

If the pseudoinverse solution (4.7) is still used in the stochastic case, it corresponds to choosing $S = \mathbb{I}$ and $\lambda \to 0^+$, and can be interpreted as the least-squares estimate of a linear mapping:

$$\mathcal{F}_Z(\cdot) = F_Z \operatorname{col}(\mathbf{u}_{\text{ini}}, \mathbf{u}, \mathbf{y}_{\text{ini}}), \tag{4.19}$$

where

$$F_Z := \operatorname*{argmin}_F \ \left\| Y_f - F \operatorname{col}(U_p, U_f, Y_p) \right\|_F^2, \tag{4.20}$$

or the solution to the least-norm problem:

$$g_{\text{pinv}} = \operatorname*{argmin}_g \ \|g\|_2^2 \qquad \text{s.t.} \quad (4.9). \tag{4.21}$$

However, this predictor fails to appropriately encode the effects of noise in the output signal matrix $Y$ with the least-squares problem (4.20) considering only the noise in $Y_f$ but not in $Y_p$.

This is known as the subspace predictor (Favoreel et al., 1999; Sedghizadeh and Beheshti, 2018; Huang et al., 2019; Fiedler and Lucia, 2021).

Another existing design proposed in Lian et al. (2023) finds the vector $g$ that minimizes the Wasserstein distance (WD) between the stochastic distribution of $\mathbf{y}_{\text{ini}}$ and that of $Y_p g$. An approximation of this objective leads to a design of

$$S = \mathbb{I}, \quad \lambda = n_y L_0 \sigma^2. \tag{4.22}$$

Section 4.3 presents the indirect data-driven predictor design using the MLE principle.

**Remark 4.3.** *The ideas in this section can be extended to include disturbances in the state-space model* (1.2)*, i.e.,*

$$\begin{cases} x_{t+1} &= Ax_t + Bu_t + Ew_t, \\ y_t &= Cx_t + Du_t + v_t, \end{cases} \tag{4.23}$$

*where $w_t \in \mathbb{R}^{n_w}$ denotes the disturbances. The disturbances $w_t$ can be treated as additional uncontrolled inputs. In many applications, the offline disturbance trajectories can be obtained retroactively. Suppose the disturbance sequence $\mathbf{w}^d := \begin{bmatrix} w_1^d & w_2^d & \dots & w_N^d \end{bmatrix}^\top$ is collected, in addition to the input-output sequence. The Hankel matrix $W$ can be constructed similar to $U$ or $Y$ in* (4.2) *by replacing $\mathbf{u}^d$ or $\mathbf{y}^d$ with $\mathbf{w}^d$. Define $\mathbf{w} := \operatorname{col}(w_{-L_0}, \dots, w_{-1}, w_0, \dots, w_{L'-1})$ as the immediate past and future disturbance sequence of length L. Then, all the results presented above can be extended by replacing $U$ with the augmented input signal matrix $\operatorname{col}(U, W)$ and $\operatorname{col}(\mathbf{u}_{\text{ini}}, \mathbf{u})$ with the augmented input sequence $\operatorname{col}(\mathbf{u}_{\text{ini}}, \mathbf{u}, \mathbf{w})$.*

## 4.2 Stochastic Data-Driven Prediction by Matrix Denoising

The problem of estimating an unknown low-rank matrix from noisy measurements has been a long-standing problem, under the name of principal component analysis (Abdi and Williams, 2010) or proper orthogonal decomposition (Berkooz et al., 1993). A well-known technique to solve this problem is the truncated singular value decomposition (TSVD), which approximates the data matrices by only keeping the most significant singular values corresponding to the true rank of the noise-free matrix. According to the Eckart-Young-Mirsky (EYM) theorem (Eckart and Young, 1936), it is the best low-rank approximation to the data in terms of both the Frobenius norm and the spectral norm. When the true rank of the underlying low-rank matrix is unknown, it is still common to rely on the EYM theorem by turning the problem into estimating the true rank, e.g., Bauer (2001) in subspace identification. The simplest method is to look for a sudden decrease in the scree plot (a plot of singular values in decreasing magnitude). Information criteria (Akaike, 1974) and cross-validation techniques (Stoica et al., 1986) are also widely used.

However, an often neglected aspect of the EYM theorem is that it only provides the optimal low-rank *approximation* to the noisy data matrix but does not guarantee any optimality of minimizing the MSE of the estimate with respect to the noise-free matrix (Nadakuditi, 2014). In other words,

matrix denoising is not the same as matrix approximation.

To solve the low-rank denoising problem, the two-step approach of first determining the rank of the estimate and then applying TSVD can be interpreted as singular value hard thresholding, where the singular values are not truncated to obtain a fixed rank but according to a specified threshold. This method has been effective in multiple matrix estimation problems (Chatterjee, 2015). Gavish and Donoho (2014) show that, for unstructured matrices, there exists an optimal choice of the hard threshold asymptotically, which is also effective with finite-dimensional matrices numerically. The hard thresholding function can be generalized to a general shrinkage function on the singular values. The asymptotically optimal shrinkage function for unstructured matrices is developed in Gavish and Donoho (2017). Data-driven and adaptive shrinkage algorithms are also proposed in Nadakuditi (2014); Josse and Sardy (2015).

Another difficulty in exploiting the low-rank prior in estimation is incorporating the structural constraints. Unlike the EYM theorem, there is no closed-form solution or convex formulation for the structured low-rank approximation (SLRA) problem (Markovsky and Rapisarda, 2008). In existing works, nonlinear optimization algorithms (Markovsky and Usevich, 2013; Park et al., 1999), iterative structural approximation (Cadzow, 1988; Li et al., 1997; Wang et al., 2019), and convex relaxation (Fazel et al., 2001; Smith, 2014) are applied to obtain locally optimal or suboptimal solutions to the problem. Structure constraints pose additional difficulties in solving the low-rank denoising problem as well. The results above for the denoising problem rely on the asymptotic distribution of the singular values of the noise matrix, which is not well-studied for most structured matrices. In this regard, Nadakuditi (2014) proposes a data-driven method to estimate the distribution from the singular values of the data matrix.

This section first reviews existing algorithms for solving the SLRA and the unstructured low-rank denoising problem. It is observed that when applied to the problem of denoising low-rank generalized Hankel matrices, these two categories of algorithms improve the standard TSVD approach from two distinct perspectives, namely enforcing structural constraints and avoiding approximating the noise matrix. Based on this observation, a novel algorithm is proposed to address the low-rank Hankel matrix denoising problem directly. It combines the data-driven singular value shrinkage approach in unstructured low-rank matrix denoising and the iterative structural approximation method in SLRA. Since rigorous statistical frameworks for low-rank Hankel matrix denoising have not been established, we focus on a numerical analysis perspective to assess the performance in terms of noise reduction by Monte Carlo simulation. It is shown numerically that, when applying to the output trajectory denoising problem described in Section 4.1.2, the proposed algorithm achieves the most significant noise reduction among all existing SLRA and low-rank denoising algorithms under different noise levels.

### 4.2.1 Structured Low-Rank Approximation

Since the true MSE depends on the unknown matrix $X$, practically, the estimation problem is usually reformulated as finding the best structured rank-$r$ approximation to the measurement $\tilde{X}$:

$$\begin{aligned} \hat{X}_{\text{SLRA}} := \quad & \underset{\hat{X}\in\mathbb{M}^{m\times n}}{\text{argmin}} \quad && \left\|\tilde{X}-\hat{X}\right\|_F^2 \\ & \text{s.t.} && \text{rank}(\hat{X}\Pi)\leq r. \end{aligned} \tag{4.24}$$

This problem will be referred to as the approximation problem. The most well-known method to solve this approximation problem is probably TSVD. Let

$$\tilde{X}\Pi = \sum_{i=1}^{m} w_i \mathbf{u}_i \mathbf{v}_i^{\mathsf{T}} \tag{4.25}$$

be the SVD of $\tilde{X}\Pi$, where $w_i$ are the singular values in decreasing magnitude and $\mathbf{u}_i \in \mathbb{R}^m$, $\mathbf{v}_i \in \mathbb{R}^n$ are the left and right singular vectors, respectively. Then, the TSVD estimate is given by

$$\hat{X}_{\text{TSVD}}(\tilde{X},\Pi;r) := \sum_{i=1}^{r} w_i \mathbf{u}_i \mathbf{v}_i^{\mathsf{T}} + \tilde{X}(\mathbb{I}_n - \Pi). \tag{4.26}$$

For the unstructured case, i.e., $\mathbb{M}^{m\times n} = \mathbb{R}^{m\times n}$, $\Pi = \mathbb{I}$, the EYM theorem (Eckart and Young, 1936) shows that $\hat{X}_{\text{TSVD}}$ is the closed-form solution to (4.24).

When the matrix is structured, closed-form solutions no longer exist in general, so relaxations or nonlinear optimization techniques are needed to solve the problem. Here, we highlight an algorithm for solving the Hankel low-rank approximation problem by iterating the TSVD step and a Hankel approximation step alternatingly. This method extends the algorithms in Wang et al. (2019); Li et al. (1997) to the generalized Hankel structure. The algorithm is outlined in Algorithm 4.2.

**Algorithm 4.2** Iterative algorithm for SLRA with generalized Hankel structure

1: **Input:** $\tilde{X}, \Pi, r, \varepsilon$
2: $\tilde{X}_1 \leftarrow \tilde{X}$
3: **repeat**
4: $\quad \tilde{X}_2 \leftarrow \hat{X}_{\text{TSVD}}(\tilde{X}_1, \Pi; r)$
5: $\quad \tilde{X}_1 \leftarrow \mathcal{H}(\tilde{X}_2)$
6: **until** $\left\|\tilde{X}_1 - \tilde{X}_2\right\| < \varepsilon \left\|\tilde{X}_1\right\|$
7: **Output:** $\hat{X} = \tilde{X}_1$

In Algorithm 4.2, $\mathcal{H}(\cdot)$ is the orthogonal projector onto the set of Hankel matrices. It can be calculated by setting all the elements along a skew diagonal to be the average value of that skew diagonal.

In addition, two other algorithms proposed in the existing literature to solve the SLRA problem are considered for the generalized Hankel structure. The first algorithm, proposed in Markovsky

and Usevich (2013), decomposes the optimization problem into a least-norm inner problem and a nonlinear outer problem and solves it by local optimization methods. The second algorithm applies the nuclear norm heuristic of the rank constraint and formulates a regularized convex optimization problem (Fazel et al., 2001):

$$\hat{X}_{\text{nuc}} := \underset{\hat{X}\in\mathbb{M}^{m\times n}}{\operatorname{argmin}} \frac{1}{2}\left\|\tilde{X}-\hat{X}\right\|_F^2 + \tau\left\|\hat{X}\Pi\right\|_*, \tag{4.27}$$

where the nuclear norm $\|\cdot\|_*$ is defined as the sum of all singular values. For the unstructured case, it has a closed-form solution of soft-thresholding the singular values:

$$\hat{X}_{\text{nuc}} = \sum_{i=1}^{m} \max(0, w_i - \tau)\mathbf{u}_i\mathbf{v}_i^{\mathsf{T}}. \tag{4.28}$$

### 4.2.2 From Approximation to Denoising

Despite its wide applications, the SLRA solution $\hat{X}_{\text{SLRA}}$ to the approximation problem does not always serve as a reasonable solution to the denoising problem. Consider an extreme case when $\sigma \to \infty$. The optimal denoising solution is a zero matrix almost surely as the low-rank matrix is overwhelmed by noise, whereas $\hat{X}_{\text{SLRA}}$ approaches infinity, giving only a low-rank approximation of the particular noise realization. This observation illustrates a critical aspect of solving the denoising problem: the noise matrix does not only contaminate the left null space of $X\Pi$ and inflate the zero singular values but also enters the column space of $X\Pi$ and inflates the non-zero singular values.

In detail, let the singular values of $X\Pi$ be $x_i$, $i = 1,\dots,m$, where $x_i = 0$ for $i > r$. Consider the asymptotic framework where $n \to \infty$ while keeping both the aspect ratio $\beta$ and the true singular values $x_i$ constant. As proved in Theorem 2.9 of Benaych-Georges and Nadakuditi (2012), the $r$ largest singular values of $\tilde{X}\Pi$ satisfy

$$\lim_{n\to\infty} w_i = \begin{cases} D_{\mu_V}^{-1}(1/x_i^2), & x_i^2 > 1/D_{\mu_V}(b^+) \\ b, & x_i^2 \le 1/D_{\mu_V}(b^+) \end{cases} \tag{4.29}$$

almost surely for $i = 1,\dots,r$, where $\mu_V$ is the asymptotic probability measure of the empirical singular value distribution of $V\Pi$:

$$\mu_V := \lim_{n\to\infty} \frac{1}{m}\sum_{i=1}^{m} \delta_{z_i}, \quad z_i\text{: singular values of } V\Pi, \tag{4.30}$$

$b$ is the supremum of the support of $\mu_V$, and $D_{\mu_V}(\cdot)$ is the D-transform under $\mu_V$ (Benaych-Georges and Nadakuditi, 2012). An important property of (4.29) is that the noisy singular values are always enlarged, i.e., $w_i > x_i$, $\forall x_i$. Therefore, in addition to setting the smallest $(m-r)$ singular values to zero, the $r$ largest singular values of $\tilde{X}\Pi$ need to be shrunk as well, depending on the singular value distribution of the noise matrix. So, for the denoising problem, it makes

sense to consider the following singular value shrinkage estimate:

$$\hat{X}_{\text{shrink}} := \sum_{i=1}^{m} \eta(w_i)\mathbf{u}_i\mathbf{v}_i^{\mathsf{T}} + \tilde{X}(\mathbb{I}_n - \Pi), \quad \eta(w_i) \in [0, w_i] \tag{4.31}$$

to counteract the effect of inflated noisy singular values.

For the unstructured case, it can be assumed that $V$ has i.i.d. unit Gaussian entries. Then, $\mu_V$ is known to follow the Marchenko-Pastur distribution (Marchenko and Pastur, 1967). In this case, it has been proven by Gavish and Donoho (2014, 2017) that the following shrinkage law obtains the minimum asymptotic MSE:

$$\eta(w) = \begin{cases} \frac{n\sigma^2}{w}\sqrt{\left(\frac{w^2}{n\sigma^2} - \beta - 1\right)^2 - 4\beta}, & w > (1+\sqrt{\beta})\sqrt{n}\sigma \\ 0, & w \le (1+\sqrt{\beta})\sqrt{n}\sigma \end{cases}. \tag{4.32}$$

In addition to the general shrinkage function (4.31), particular shrinkage functions with piecewise linear forms are often considered. These include hard thresholding and soft thresholding functions, which are defined as

$$\eta_H(w) = w\mathbf{1}_{\{w \ge \tau_H\}}, \quad \eta_S(w) = \max(0, w - \tau_S), \tag{4.33}$$

respectively. These functions correspond to TSVD with rank estimation and nuclear norm regularization for the unstructured case. The optimal thresholds are

$$\tau_H = \sqrt{2(\beta+1) + \frac{8\beta}{\beta + 1 + \sqrt{\beta^2 + 14\beta + 1}}}\sigma\sqrt{n}, \tag{4.34}$$

$$\tau_S = (1+\sqrt{\beta})\sigma\sqrt{n}, \tag{4.35}$$

respectively. Interestingly, these asymptotically optimal results do not require knowledge of the true rank $r$. For a comparison of these shrinkage functions, see Figure 2 in Gavish and Donoho (2017).

When the noise level $\sigma$ is unknown, it can be estimated by comparing the last $(m-r)$ singular values of $\tilde{X}\Pi$, which are dominated by noise, to the theoretical Marchenko-Pastur distribution. In this work, we apply a robust and consistent estimator proposed in Section III.E of Gavish and Donoho (2014):

$$\hat{\sigma} := \frac{w_{\text{med}}}{\sqrt{n \cdot z_{\text{med}}(\beta)}}, \tag{4.36}$$

where $w_{\text{med}}$ is the median singular value and $z_{\text{med}}(\beta)$ is the median of the Marchenko-Pastur distribution solved by the equation

$$\int_{(1-\sqrt{\beta})^2}^{z_{\text{med}}(\beta)} \frac{\sqrt{\left((1+\sqrt{\beta})^2 - t\right)\left(t - (1-\sqrt{\beta})^2\right)}}{2\pi t}\mathrm{d}t = \frac{1}{2}. \tag{4.37}$$

### 4.2.3 Denoising with Generalized Hankel Structure

When $V$ is Hankel, the assumption of i.i.d. Gaussian entries in the previous subsection is violated. If the probability measure $\mu_V$ for a Hankel $V$ is known, the optimal shrinkage law (4.32) can be generalized as

$$\eta(w;\mu_V) = \begin{cases} -2\frac{D_{\mu_V}(w)}{D'_{\mu_V}(w)}, & D_{\mu_V}(w) < D_{\mu_V}(b^+) \\ 0, & D_{\mu_V}(w) \geq D_{\mu_V}(b^+) \end{cases}, \tag{4.38}$$

according to Theorem 2.1 in Nadakuditi (2014). Unfortunately, to the best of our knowledge, the empirical singular value distribution of random Hankel matrices has only been analyzed numerically (e.g., Ghodsi et al. (2015); Smith (2014)) but lacks an analytical formulation.

So, instead of aiming to derive the optimal shrinkage law analytically, the data-driven singular value shrinkage algorithm, *OptShrink*, can be applied (Algorithm 1 in Nadakuditi (2014)). This algorithm obtains a consistent estimate of the noise singular value distribution from the last $(m-r)$ singular values of $\tilde{X}\Pi$. It can be considered as an extension of the noise level estimator (4.36), where the distribution has been parametrized by noise level $\hat{\sigma}$ with the Marchenko-Pastur distribution for the unstructured case. Here, a nonparametric estimation of the empirical singular value distribution is obtained. So, the data-driven shrinkage law is given by

$$\eta_{\mathrm{DD}}(w_i) = \begin{cases} \eta(w_i;\hat{\mu}_V(w_{r+1},\ldots,w_m)), & i = 1,\ldots,r \\ 0, & i = r+1,\ldots,m \end{cases}. \tag{4.39}$$

Note that this algorithm requires knowledge of the true rank $r$ to distinguish the singular values resulting only from noise. When the true rank is unavailable, it can be replaced with an upper rank bound $\hat{r} \geq r$.

In addition to the problem with the generalized Hankel noise model, the previous algorithms to solve the denoising problem also do not guarantee the Hankel structure of the unknown matrix $X$. To enforce the Hankel structure in the denoised estimate, we modify Algorithm 4.2 by replacing the TSVD solution with the data-driven singular value shrinkage law (4.39) as follows:

**Algorithm 4.3** Iterative algorithm for low-rank denoising with generalized Hankel structure

1: **Input:** $\tilde{X}, \Pi, r, \varepsilon$
2: $\tilde{X}_1 \leftarrow \tilde{X}$
3: **repeat**
4: $\quad \tilde{X}_2 \leftarrow \sum_{i=1}^{r} \eta(w_i;\hat{\mu}_V(w_{r+1},\ldots,w_m))\mathbf{u}_i\mathbf{v}_i^{\mathsf{T}} + \tilde{X}_1(\mathbb{I}_n - \Pi)$, where $(w_i)_{i=1}^m$ are the singular values of $\tilde{X}_1\Pi$.
5: $\quad \tilde{X}_1 \leftarrow \mathcal{H}(\tilde{X}_2)$
6: **until** $\left\|\tilde{X}_1 - \tilde{X}_2\right\| < \varepsilon \left\|\tilde{X}_1\right\|$
7: **Output:** $\hat{X} = \tilde{X}_1$

### 4.2.4 Numerical Results

The performance of the algorithms discussed in the previous subsections is compared numerically on the output trajectory denoising problem discussed in Section 4.1.2 by Monte Carlo simulation. The algorithms are listed as follows.

1. Truncated singular value decomposition (*TSVD*): Equation (4.26)
2. Structured low-rank approximation by iteration (*Iter*): Algorithm 4.2 with $\varepsilon = 10^{-5}$
3. Structured low-rank approximation by local optimization (*SLRA*): the SLRA package (Markovsky and Usevich, 2014)
4. Nuclear norm regularization (*Nuc*): convex optimization problem (4.27) with $\tau$ selected as the optimal soft threshold (4.35)
5. Optimal shrinkage law (*Shrink*): Equation (4.31) with shrinkage law (4.32)
6. Optimal hard thresholding (*Hard*): Equation (4.31) with hard threshold (4.34)
7. Data-driven shrinkage law (*DD*): Equation (4.31) with data-driven shrinkage law (4.39)
8. Iterative low-rank Hankel matrix denoising (*LRHD*): Algorithm 4.3 with $\varepsilon = 10^{-5}$

Among these algorithms, (2)–(4) are SLRA methods, (5)–(7) are unstructured matrix denoising methods, and (8) is the proposed method. When needed, the true rank $r$ is assumed to be known.

Random fourth-order systems generated by the `drss` function in MATLAB are considered ($n_x = r = 4$) in Monte Carlo simulation. The number of rows is selected as $m = L = 8$. The additive noise $v_t$ is considered as i.i.d. Gaussian noise with $\mathcal{N}(0, \sigma^2)$. Two different noise levels of $\sigma^2 = 0.1$ and $0.01$ are considered. The noise level $\sigma$ is assumed unknown for the algorithms. The trajectory length is selected as $N = 96$.

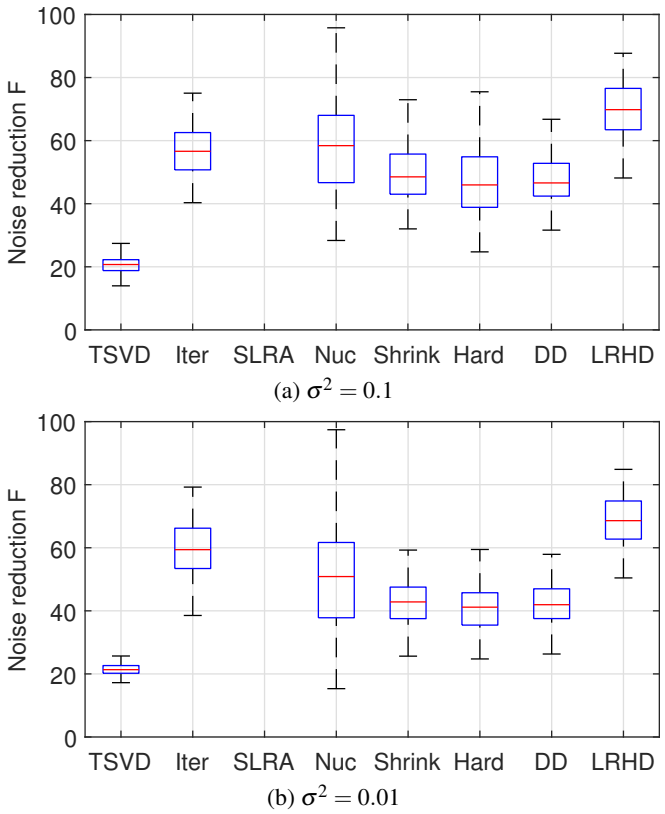
The performance is assessed by the following noise reduction measure:

$$F := 100 \cdot \left(1 - \frac{\left\|X - \hat{X}\right\|_F}{\left\|X - \tilde{X}\right\|_F}\right), \tag{4.40}$$

where $F = 0$ means no noise reduction and $F = 100$ means the noise-free matrix is fully recovered. For each test case, 100 Monte Carlo simulations are conducted.

The boxplots of the noise reduction measure $F$ are plotted in Figures 4.1. It can be seen that the proposed iterative low-rank Hankel matrix denoising algorithm achieves the most significant noise reduction at both noise levels. This result proves the benefit of combining the asymptotically optimal singular value shrinkage law with structural constraints. For the most part, the other SLRA and matrix denoising algorithms also perform much better than the TSVD approach, except that *SLRA* fails to obtain a reasonable solution. This demonstrates that despite being the optimal low-rank approximation for unstructured matrices, the performance of TSVD is not satisfying in terms of denoising the structured low-rank matrix.

Similar simulations are also conducted for the impulse response denoising problem described in




(a) $\sigma^2 = 0.1$

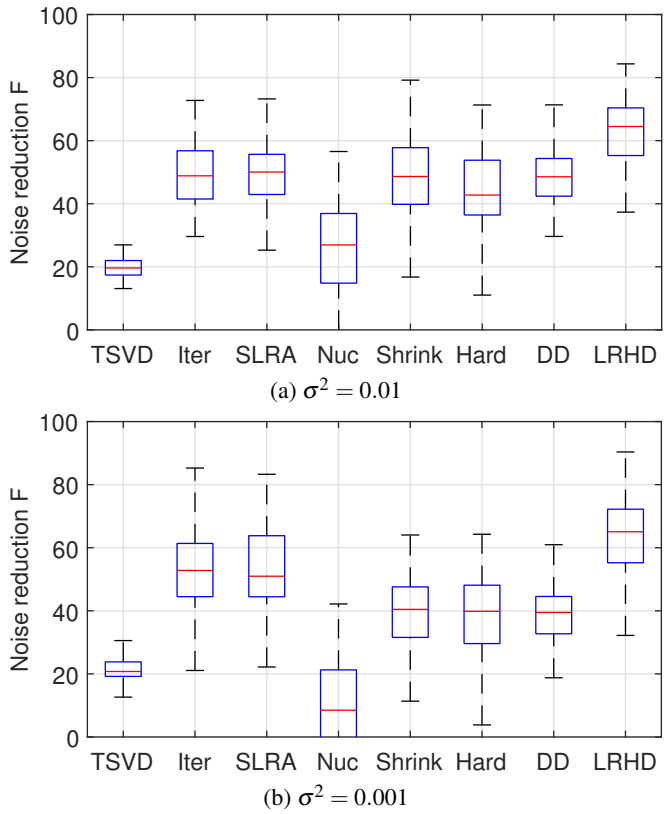
(b) $\sigma^2 = 0.01$

Figure 4.1: Noise reduction performance for the output trajectory denoising problem.

Remark 4.2 with a shorter length of $N = 40$ since the impulse response decays exponentially for stable systems. Two different noise levels of $\sigma^2 = 0.01$ and $0.001$ are considered. Similar results are observed in Figure 4.2, except that *SLRA* works, whereas *Nuc* does not perform well in this example.

## 4.3 Maximum Likelihood Prediction: the Signal Matrix Model

This section presents a tuning-free and well-defined stochastic data-driven predictor by maximum likelihood estimation (MLE). Since this predictor is expressed directly in terms of the signal matrix, we name it the signal matrix model (SMM). For simplicity of exposition, the results in the section are stated for the SISO case, but they seamlessly hold for the MIMO case. The derivation and the computation of the predictor are proposed in Sections 4.3.1 and 4.3.2, respectively, followed by discussions on preconditioning for large datasets and performance analysis in Sections 4.3.3 and 4.3.4, respectively. The predictor is applied to impulse response estimation in Section 4.3.5.




(a) $\sigma^2 = 0.01$

(b) $\sigma^2 = 0.001$

Figure 4.2: Noise reduction performance for the impulse response denoising problem.

### 4.3.1 Derivation of the Maximum Likelihood Estimator

To develop the maximum likelihood predictor, we first need to analyze how the output noise propagates in the predictor. By considering zero-mean i.i.d. Gaussian output noise, the distributions of $\mathbf{y}_{\text{ini}}$ and $Y$ are also Gaussian. In what follows, the distributions are denoted by

$$\mathbf{y}_{\text{ini}} \sim \mathcal{N}\left(\mathbf{y}_{\text{ini}}^0, \Sigma_{\text{yini}}\right), \quad \text{vec}\,(Y) \sim \mathcal{N}\left(\text{vec}\left(Y^0\right), \Sigma_Y\right), \tag{4.41}$$

where $\mathbf{y}_{\text{ini}}^0$ and $Y^0 =: \text{col}\left(Y_p^0, Y_f^0\right)$ are noise-free versions of $\mathbf{y}_{\text{ini}}$ and $Y$, respectively, and $\mathbf{y}_{\text{ini}}$ is uncorrelated with $Y$. Then, for a given $g$, the distribution of $Yg = \left(g^\mathsf{T} \otimes \mathbb{I}_L\right) \text{vec}\,(Y)$ is

$$Yg|\, g \sim \mathcal{N}\left(Y^0 g, \underbrace{\begin{bmatrix} \Sigma_p & \Sigma_{pf} \\ \Sigma_{pf}^\mathsf{T} & \Sigma_f \end{bmatrix}}_{\Sigma_g}\right), \tag{4.42}$$

where

$$\Sigma_g := \left(g^\mathsf{T} \otimes \mathbb{I}_L\right) \Sigma_Y \left(g \otimes \mathbb{I}_L\right). \tag{4.43}$$

Here, the property of the Kronecker product $\text{vec}(ABC) = (C^\mathsf{T} \otimes A)\text{vec}(B)$ is used.

Specifically, consider the following noise model:

$$\begin{cases} y_i^d = y_i^{d,0} + v_i^d, & v_i^d \sim \mathcal{N}(0, \sigma^2), \\ \mathbf{y}_{\text{ini}} = \mathbf{y}_{\text{ini}}^0 + \mathbf{v}_p, & \mathbf{v}_p \sim \mathcal{N}(\mathbf{0}, \sigma_p^2 \mathbb{I}). \end{cases} \tag{4.44}$$

The propagated covariance matrix $\Sigma_g$ is derived in the following lemma.

**Lemma 4.1.** *When $Y$ is constructed with the Hankel structure* (4.2)*, $(\Sigma_g)_{i,j} = \sigma^2 R_{gg}(i-j)$, where*

$$R_{gg}(\tau) := \sum_{k=1}^{M-|\tau|} g_k g_{k+|\tau|} \tag{4.45}$$

*is the sample autocorrelation of $g$ without normalization and $g_k$ denotes the $k$-th entry of $g$.*

*Proof.* According to the noise model of $y_i^d$ and the Hankel structure of $Y$, $\Sigma_Y$ can be expressed as

$$(\Sigma_Y)_{i,j} = \begin{cases} \sigma^2, & (\text{vec}(Y))_i = (\text{vec}(Y))_j \\ 0, & \text{otherwise} \end{cases}. \tag{4.46}$$

Let $\zeta_i \in \mathbb{R}^L$ be the $i$-th column of $\left(g^\mathsf{T} \otimes \mathbb{I}_L\right)$, and $\mathcal{S} := \left\{ (i,j) \,\middle|\, (\text{vec}(Y))_i = (\text{vec}(Y))_j \right\}$. We have

$$\Sigma_g = \sigma^2 \sum_{(i,j)\in\mathcal{S}} \zeta_i \zeta_j^\mathsf{T}. \tag{4.47}$$

Let the $i$-th and the $j$-th entries of $\text{vec}(Y)$ 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 $\left(g^\mathsf{T} \otimes \mathbb{I}_L\right)$, 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 $q$-th standard basis vector, and similarly for $\mathbf{e}_s$. Thus,

$$\Sigma_g = \sigma^2 \sum_{q+r=s+t} g_r g_t \mathbf{e}_q \mathbf{e}_s^\mathsf{T}. \tag{4.48}$$

So the $(q,s)$-th entry of $\Sigma_g$ is given by

$$(\Sigma_g)_{q,s} = \sigma^2 \sum_{q+r=s+t} g_r g_t, \tag{4.49}$$

which directly leads to Lemma 4.1. □

**Remark 4.4** (Data-Driven Noise Level Estimation)**.** *When the noise level $\sigma^2$ in the output signal matrix $Y$ is unknown, it can be estimated using the same method* (4.36) *as discussed in Section 4.2.2. It is applicable when the median singular value $w_{\text{med}}$ comes purely from noise, i.e., $n_y L > 2n_x$. The noise level of online data $\sigma_p^2$ can be set to zero when initial conditions are known exactly or to $\sigma^2$ when the same sensor is used for offline and online measurements. Otherwise, online measurements can be taken beforehand, and $\sigma_p^2$ can be estimated similarly to $\sigma^2$.*

Define

$$\tilde{\mathbf{y}} := \begin{bmatrix} \varepsilon_y \\ \hat{\mathbf{y}} \end{bmatrix} = Yg - \begin{bmatrix} \mathbf{y}_{\text{ini}} \\ \mathbf{0} \end{bmatrix} = \left( g^{\mathsf{T}} \otimes \mathbb{I}_L \right) \text{vec}(Y) - \begin{bmatrix} \mathbf{y}_{\text{ini}} \\ \mathbf{0} \end{bmatrix}, \tag{4.50}$$

where $\varepsilon_y := Y_p g - \mathbf{y}_{\text{ini}}$ is the residual of the past output relation, representing the total deviation of the past output trajectory from the noise-free case. Then, we want to construct an estimator that maximizes the conditional probability of observing the realization $\tilde{\mathbf{y}}$ corresponding to the available data given $g$. The statistics of $\tilde{\mathbf{y}}$ given $g$ are given by

$$\mathbb{E}(\tilde{\mathbf{y}}|g) = \mathbb{E}(Y)g - \begin{bmatrix} \mathbb{E}(\mathbf{y}_{\text{ini}}) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} Y_p^0 g - \mathbf{y}_{\text{ini}}^0 \\ Y_f^0 g \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ Y_f^0 g \end{bmatrix}, \quad \text{cov}(\tilde{\mathbf{y}}|g) = \Sigma_g + \begin{bmatrix} \sigma_p^2 \mathbb{I} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} \end{bmatrix} =: \tilde{\Sigma}_g. \tag{4.51}$$

Thus, due to the linearity of the normal distribution, we have

$$\tilde{\mathbf{y}}|g \sim \mathcal{N}\left( \begin{bmatrix} \mathbf{0} \\ Y_f^0 g \end{bmatrix}, \tilde{\Sigma}_g \right), \tag{4.52}$$

which has the probability density

$$p(\tilde{\mathbf{y}}|g) = (2\pi)^{-\frac{L}{2}} \det(\tilde{\Sigma}_g)^{-\frac{1}{2}} \exp\left( -\frac{1}{2} \begin{bmatrix} Y_p g - \mathbf{y}_{\text{ini}} \\ Y_f g - Y_f^0 g \end{bmatrix}^{\mathsf{T}} \tilde{\Sigma}_g^{-1} \begin{bmatrix} Y_p g - \mathbf{y}_{\text{ini}} \\ Y_f g - Y_f^0 g \end{bmatrix} \right). \tag{4.53}$$

Note that here $Y_f^0$ is also unknown and can be estimated with the maximum likelihood approach as well. In this way, we are ready to derive the signal matrix model (SMM) by solving the following optimization problem:

$$\min_{g \in \mathcal{G}, Y_f^0} \quad -\log p(\tilde{\mathbf{y}}|g, Y_f^0), \tag{4.54}$$

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

Substituting (4.53) into (4.54), we have the equivalent optimization problem:

$$\min_{g \in \mathcal{G}, Y_f^0} \quad \text{logdet}(\tilde{\Sigma}_g(g)) + \begin{bmatrix} Y_p g - \mathbf{y}_{\text{ini}} \\ Y_f g - Y_f^0 g \end{bmatrix}^{\mathsf{T}} \tilde{\Sigma}_g^{-1}(g) \begin{bmatrix} Y_p g - \mathbf{y}_{\text{ini}} \\ Y_f g - Y_f^0 g \end{bmatrix}. \tag{4.55}$$

It is easy to see that the optimal value of $Y_f^0$ is $Y_f$ regardless of the choice of $g$. So (4.55) is equivalent to

$$\min_{g \in \mathcal{G}} \quad \text{logdet}(\tilde{\Sigma}_g(g)) + \begin{bmatrix} Y_p g - \mathbf{y}_{\text{ini}} \\ \mathbf{0} \end{bmatrix}^{\mathsf{T}} \tilde{\Sigma}_g^{-1}(g) \begin{bmatrix} Y_p g - \mathbf{y}_{\text{ini}} \\ \mathbf{0} \end{bmatrix}. \tag{4.56}$$

In this objective function, the first term indicates the uncertainty of the prediction, whereas the second term penalizes the deviation from the past output measurements.

### 4.3.2 Iterative Computation of the Estimator

Unfortunately, (4.56) is a non-convex problem. To find a computationally efficient algorithm to solve (4.56), we relax the problem and solve it with sequential quadratic programming (SQP) (Boggs and Tolle, 1995). First, the cross-correlation between elements in $Y$ is neglected and the covariance matrix $\tilde{\Sigma}_g$ is approximated with its diagonal part, denoted by $\bar{\Sigma}_g$, i.e.,

$$\left(\bar{\Sigma}_g\right)_{i,j} := \begin{cases} \left(\tilde{\Sigma}_g\right)_{i,j}, & i = j \\ 0, & i \neq j \end{cases}. \tag{4.57}$$

**Remark 4.5.** *When the signal matrix Z is constructed as a Page matrix or from independent trajectories as discussed in Remark 4.1, it is easy to see that $\Sigma_g$ is diagonal with $\Sigma_g = \sigma^2 \|g\|_2^2 \mathbb{I}$ and this approximation holds exactly.*

**Remark 4.6.** *This approximation gives an upper bound on the log-det terms. According to Hadamard's inequality, since $\tilde{\Sigma}_g \in \mathbb{S}_{++}^L$, we have* $\operatorname{logdet}(\tilde{\Sigma}_g(g)) \leq \operatorname{logdet}(\bar{\Sigma}_g(g))$.

In this way, problem (4.56) is approximated as

$$\min_{g \in \mathcal{G}} \quad L' \log\left(\|g\|_2^2\right) + L_0 \log\left(\sigma^2 \|g\|_2^2 + \sigma_p^2\right) + \frac{1}{\sigma^2 \|g\|_2^2 + \sigma_p^2} \|Y_p g - \mathbf{y}_{\text{ini}}\|_2^2. \tag{4.58}$$

This problem can be readily solved by SQP. The following QP problem is solved for each iteration.

$$\begin{aligned} g^{(k+1)} = \quad & \underset{g}{\operatorname{argmin}} \quad \lambda(g^{(k)}) \|g\|_2^2 + \|Y_p g - \mathbf{y}_{\text{ini}}\|_2^2 \\ & \text{s.t.} \qquad \begin{bmatrix} U_p \\ U_f \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{ini}} \\ \mathbf{u} \end{bmatrix}, \end{aligned} \tag{4.59}$$

where $\lambda(g^{(k)}) := L'\sigma_p^2 \Big/ \left\|g^{(k)}\right\|_2^2 + L\sigma^2$. The objective function in (4.58) 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 solution to (4.59) is given in (4.16) with $S = \mathbb{I}$ and $\lambda = \lambda(g^{(k)})$ and denoted as

$$g^{(k+1)} = R_1(g^{(k)})\,\tilde{\mathbf{u}} + R_2(g^{(k)})\,\mathbf{y}_{\text{ini}} \tag{4.60}$$

with slight abuse of notation. This algorithm converges to a local minimum of the problem (4.58).

**Remark 4.7.** *The formulation can be straightforwardly extended to other noise models, including correlated noise, input noise, and alternative noise distributions. For example, when the noise is Laplacian, it would lead to an $l_1$-norm penalization in the estimator similar to the regularizer proposed in Coulson et al. (2019); when i.i.d. Gaussian input errors also exist in offline and online data, an additional input regularization term $\|Ug - \tilde{\mathbf{u}}\|_2^2$ would occur in the iterative algorithm, i.e., $g^{(k+1)} = \underset{g}{\operatorname{argmin}}\ \|g\|_2^2 + \lambda_1(g^{(k)}) \|Y_p g - \mathbf{y}_{\text{ini}}\|_2^2 + \lambda_2(g^{(k)}) \|Ug - \tilde{\mathbf{u}}\|_2^2$.*

Based on the derived maximum likelihood estimator of $g$, the step of solving the linear system

(4.9) in Algorithm 4.1 can be replaced by solving the SQP problem (4.59). The SQP problem can be initialized at the pseudoinverse solution $g_{\text{pinv}}$. This leads to Algorithm 4.4 for maximum likelihood data-driven prediction.

**Algorithm 4.4** Maximum likelihood data-driven prediction: the signal matrix model (SMM)

1: **Given:** signal matrix $Z$, noise parameters $\sigma, \sigma_p$, $\varepsilon$
2: **Input:** $\mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}, \mathbf{u}$.
3: $k \leftarrow 0$, $g^{(0)} \leftarrow g_{\text{pinv}}$ from (4.8)
4: **repeat**
5: Calculate $g^{(k+1)}$ with (4.60).
6: $k \leftarrow k+1$
7: **until** $\left\|g^{(k)} - g^{(k-1)}\right\| < \varepsilon \left\|g^{(k-1)}\right\|$
8: **Output:** $g_{\text{SMM}} = g^{(k)}$, $\hat{\mathbf{y}} = Y_f g_{\text{SMM}}$

### 4.3.3 Preconditioning of the Signal Matrix

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 intermediate 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, at most $2L$ independent basis vectors are needed to describe all the possible input-output trajectories of length $L$. Therefore, it is possible to precondition the signal matrix such that only $2L$ basis trajectories are used.

To do this, we propose the following strategy based on SVD to compress the data such that the dimension of the parameter vector $g$ is $2L$ regardless of the raw data length. Let $Z = \Omega S V^\mathsf{T} \in \mathbb{R}^{2L \times M}$ be the SVD of the signal matrix. Define the compressed signal matrix

$$\tilde{Z} := \operatorname{col}\left(\tilde{U}_p, \tilde{U}_f, \tilde{Y}_p, \tilde{Y}_f\right) := \Omega S_{2L} \in \mathbb{R}^{2L \times 2L}, \tag{4.61}$$

where $\tilde{U}_p, \tilde{Y}_p \in \mathbb{R}^{L_0 \times 2L}$, $\tilde{U}_f, \tilde{Y}_f \in \mathbb{R}^{L' \times 2L}$, and $S_{2L}$ is the first $2L$ columns of $S$.

It is shown in the following proposition that Algorithm 4.4 with the compressed signal matrix obtains the same output trajectory $\hat{\mathbf{y}}$ as with the raw signal matrix.

**Proposition 4.1.** *Let the predicted trajectories with signal matrices $Z$ and $\tilde{Z}$ from Algorithm 4.4 be $\hat{\mathbf{y}}$ and $\hat{\mathbf{y}}'$, respectively. Then we have $\hat{\mathbf{y}}' = \hat{\mathbf{y}}$.*

*Proof.* Define the transformed signal matrix $\bar{Z} := \operatorname{col}\left(\bar{U}_p, \bar{U}_f, \bar{Y}_p, \bar{Y}_f\right) := \Omega S$. Then the relations between the signal matrices are given by $Z = \tilde{Z} V_{2L}^\mathsf{T}$, $Z = \bar{Z} V^\mathsf{T}$, and $\bar{Z} = \begin{bmatrix} \tilde{Z} & \mathbb{0} \end{bmatrix}$, where $V_{2L}$ denotes the first $2L$ columns of $V$.

Denote the variables with the compressed signal matrix by a tilde and the variables with the

transformed signal matrix 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 $g^{(k+1)} = V\bar{g}^{(k+1)}$, and $\bar{g}^{(k+1)} = \text{col}\left(\tilde{g}^{(k+1)}, \mathbf{0}\right)$. This leads to $g^{(k+1)} = V_{2L}\,\tilde{g}^{(k+1)}$ and $\|g^{(k+1)}\|_2^2 = \|\tilde{g}^{(k+1)}\|_2^2$. Thus for all $k$, we have $g^{(k)} = V_{2L}\,\tilde{g}^{(k)}$ by induction. Therefore, the predicted trajectory satisfies $\hat{\mathbf{y}} = Y_f g^{(k)} = \tilde{Y}_f \tilde{g}^{(k)} = \hat{\mathbf{y}}'$. □

**Remark 4.8.** *It can be seen from the proof that $\bar{\Sigma}_g(g) = \bar{\Sigma}_g(\tilde{g})$. So, with the compressed signal matrix, the output trajectory estimate has the same covariance as the raw signal matrix when Page matrices are used and the same diagonal components when Hankel matrices are used.*

Proposition 4.1 shows that regardless of the size of the dataset, an SVD operation can be conducted offline to reduce the size of the signal matrix to a square matrix while obtaining the same results as with the original signal matrix. In this way, the online computational complexity only depends on the prediction horizon.

### 4.3.4 Comparison of Data-Driven Predictors

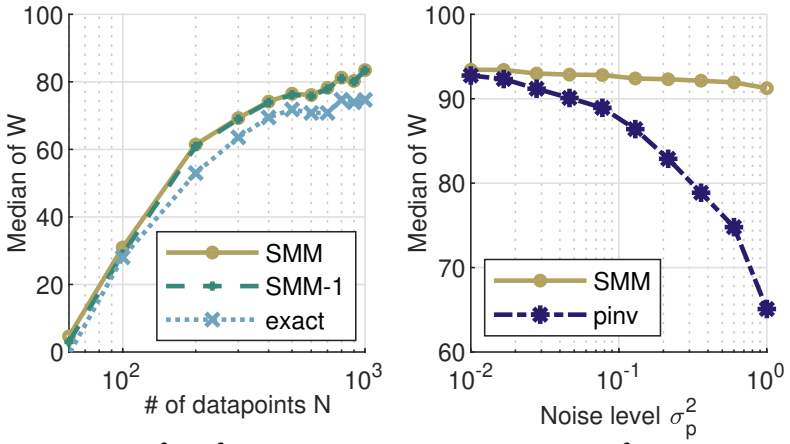
The performance of Algorithm 4.4 is analyzed numerically by comparing the accuracy of the predicted output $\hat{\mathbf{y}}$ measured by the fitting metric similar to (1.8):

$$W := 100 \cdot \left(1 - \left[\frac{\sum_{i=1}^{L'}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{L'}(y_i - \bar{y})^2}\right]^{1/2}\right), \tag{4.62}$$

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 (4.8), 2) *exact*: the SQP solution of the exact MLE problem (4.56) initialized at $g_{\text{pinv}}$, 3) *SMM-1*: the solution after one iteration of Algorithm 4.4, and 4) *SMM*: Algorithm 4.4.

Consider random SISO systems with state dimensions between 2 and 10 (generated by MATLAB function `drss`). The following parameters are used: $L_0 = n_x$ and $L' = 10$. Inputs for the identification data $(u_i^d)_{i=1}^N$ and simulation conditions $\mathbf{u}_{\text{ini}}$, $\mathbf{y}_{\text{ini}}$, $\mathbf{u}$ are all i.i.d. unit Gaussian. For each analysis, 100 Monte Carlo simulations are conducted.

The prediction accuracy of different MLE algorithms are plotted in Figure 4.3(a) for different data sizes $N$. For small data sizes, the *exact* estimate obtains very similar performance to the SMM estimates. This indicates that the approximate solution closely matches the original MLE problem. Due to the increasing dimension of $g$, the performance of *exact*, where the data compression scheme does not apply, becomes worse for larger data sizes. On the other hand, Algorithm 4.4 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) $N = 100$, $\sigma^2 = 0.01$

Figure 4.3: Comparison of prediction accuracy with different data-driven predictors.

The *SMM* estimate is compared against *pinv* in Figure 4.3(b) for different online noise levels $\sigma_p^2$. 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 $\sigma_p^2$ is large. To assess the general validity of the results shown in Figure 4.3(b), it is demonstrated theoretically in the following proposition that the SMM obtains a smaller covariance than the least-norm solution when noise is present only in $\mathbf{y}_{\text{ini}}$.

**Proposition 4.2.** *Let $g_{\text{pinv}}$ and $g_{\text{SMM}}$ be the estimates from the least-norm solution (4.8) and Algorithm 4.4, respectively. When $\sigma^2 = 0$, we have* $\operatorname{tr}(\operatorname{cov}(g_{\text{SMM}})) < \operatorname{tr}(\operatorname{cov}(g_{\text{pinv}}))$.

*Proof.* Let $K_\lambda := F^{-1} - F^{-1}U^\mathsf{T}(UF^{-1}U^\mathsf{T})^{-1}UF^{-1}$ and $g_\lambda := K_\lambda Y_p^\mathsf{T}\mathbf{y}_{\text{ini}} + R_1\tilde{\mathbf{u}}$. As discussed in Section 4.1.2, when $\lambda \to 0^+$, $g_\lambda$ converges to $g_{\text{pinv}}$. When $\lambda = L'\sigma_p^2/\|g_\lambda\|_2^2$, $g_\lambda = g_{\text{SMM}}$ since $\sigma^2 = 0$. Then we have $\operatorname{cov}(g_\lambda) = \sigma_p^2(K_\lambda Y_p^\mathsf{T})(K_\lambda Y_p^\mathsf{T})^\mathsf{T}$. The derivative of $\operatorname{tr}(\operatorname{cov}(g_\lambda))$ with respect to $\lambda$ is calculated as follows:

$$\begin{aligned}\frac{\partial \operatorname{tr}(\operatorname{cov}(g_\lambda))}{\partial (F^{-1})_{i,j}} &= \operatorname{tr}\left[\left(\frac{\partial \operatorname{tr}(\operatorname{cov}(g_\lambda))}{\partial K_\lambda}\right)^\mathsf{T}\frac{\partial K_\lambda}{\partial (F^{-1})_{i,j}}\right] \\ &= 2\sigma_p^2 \operatorname{tr}\left[\left(Y_p^\mathsf{T}Y_pK_\lambda\right)^\mathsf{T}K_\lambda F\Delta(i,j)FK_\lambda\right],\end{aligned} \tag{4.63}$$

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,

$$\begin{aligned}\frac{\partial \operatorname{tr}(\operatorname{cov}(g_\lambda))}{\partial \lambda} &= \operatorname{tr}\left[\left(\frac{\partial \operatorname{tr}(\operatorname{cov}(g_\lambda))}{\partial F^{-1}}\right)^\mathsf{T}\frac{\partial F^{-1}}{\partial \lambda}\right] \\ &= -2\sigma_p^2 \operatorname{tr}\left[\left(FK_\lambda\left(Y_p^\mathsf{T}Y_pK_\lambda\right)^\mathsf{T}K_\lambda F\right)^\mathsf{T}F^{-2}\right] \\ &= -2\sigma_p^2 \operatorname{tr}\left(K_\lambda Y_p^\mathsf{T}Y_pK_\lambda K_\lambda\right).\end{aligned} \tag{4.64}$$

According to the Schur complement, since

$$\begin{bmatrix} F^{-1} & F^{-1}U^{\mathsf{T}} \\ UF^{-1} & UF^{-1}U^{\mathsf{T}} \end{bmatrix} = \begin{bmatrix} \mathbb{I} \\ U \end{bmatrix} F^{-1} \begin{bmatrix} \mathbb{I} & U^{T} \end{bmatrix} \succ 0, \tag{4.65}$$

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 $\lambda$. This directly leads to Proposition 4.2. □

### 4.3.5 Impulse Response Estimation as Trajectory Prediction Problem

One direct application of the derived stochastic data-driven trajectory predictor in system identification is to identify the impulse responses of the system by simulating the SMM with a pulse input. Numerical tests show that model fitting is improved compared to the conventional least-squares estimate when the truncation error is significant, or the input history is unknown.

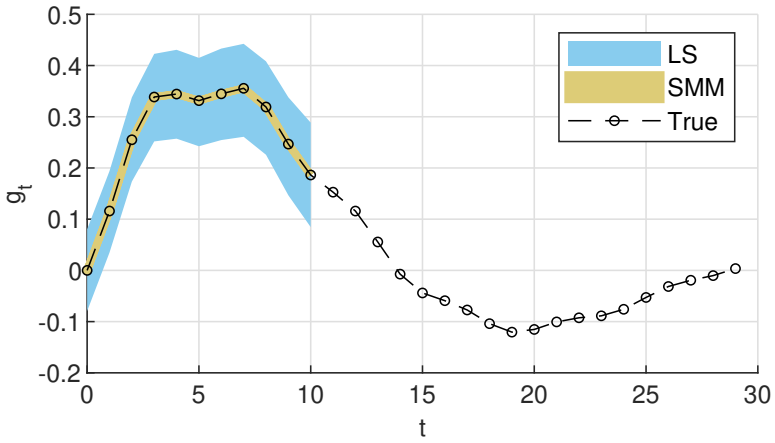
Consider the regression problem (3.2) for estimating the impulse response discussed in Section 3.1. Note that there is an overload of the notation $g$ since both are the conventional notation used in the literature. There are two main assumptions underlying this formulation and the associated least-squares estimate (3.3): 1) the truncation error of the finite impulse response is negligible, i.e., $g_i \approx 0$ for all $i \geq n_g$, and 2) data collection starts as rest, i.e., $u_t^d = 0$, $\forall t \leq 0$. Otherwise additional input measurements $(u_i^d)_{i=2-n_g}^{0}$ are required or the first $n_g - 1$ rows need to be discarded.

However, these assumptions may not be satisfied in practice. When the state 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. The least-squares algorithm may become impractical in this case due to the limited data length. 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 $(n_g - 1)$ measurements cannot be used, under which case the data efficiency is substantially affected when a large $n_g$ is needed.

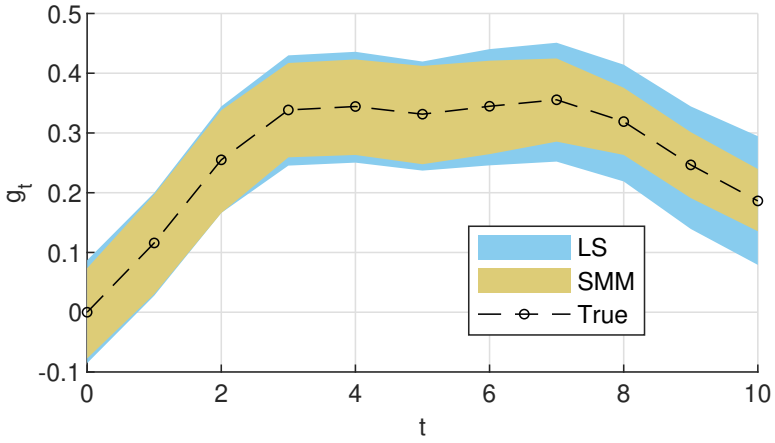
Instead, we propose using the SMM to estimate the impulse response by finding the length-$n_g$ response to a pulse input from zero initial conditions, i.e.,

$$\mathbf{u}_{\text{ini}} = \mathbf{0}, \mathbf{y}_{\text{ini}} = \mathbf{0}, \mathbf{u} = \operatorname{col}(1, \mathbf{0}), L' = n_g. \tag{4.66}$$

Since the initial condition is known exactly, we have $\sigma_p = 0$. Then the output trajectory $\hat{\mathbf{y}}$ is an estimate of the length-$n_g$ impulse response of the system (Markovsky et al., 2005b). This approach requires neither of the assumptions for the least-squares method. Instead of requiring a length-$(n_g - 1)$ 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 $n_g$ and unknown input history as shown in Theorem 4.1.3, as long as the persistency of excitation condition is satisfied.




(a) Noise-free case




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

Figure 4.4: Comparison of impulse response estimation with truncation errors. Colored area: estimates within two standard deviations.

The SMM-based algorithm is tested against the least-squares estimate by applying it to numerical examples. We compare the proposed estimate *SMM* (Algorithm 4.4 with (4.66)) with the least-squares estimate *LS* (3.3). The parameters used in the simulation are $N = 50$, $L_0 = 4$, $n_g = L' = 11$, $\sigma^2 = 0.01$. In *SMM*, the noise level $\sigma^2$ is estimated using (4.36). The identification data are generated with i.i.d. unit Gaussian input signals. For each case, 1000 Monte Carlo simulations are conducted.

In the first example, we consider the fourth-order LTI system $G_2(q)$ defined in (3.26). This system is relatively slow. The truncation error is significant when $n_g = 11$ is selected. First, the *LS* and *SMM* algorithms are compared in the noise-free case, and the results are shown in Figure 4.4(a). It can be 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 4.4(b). The *SMM* estimator has a smaller variance compared to *LS*.

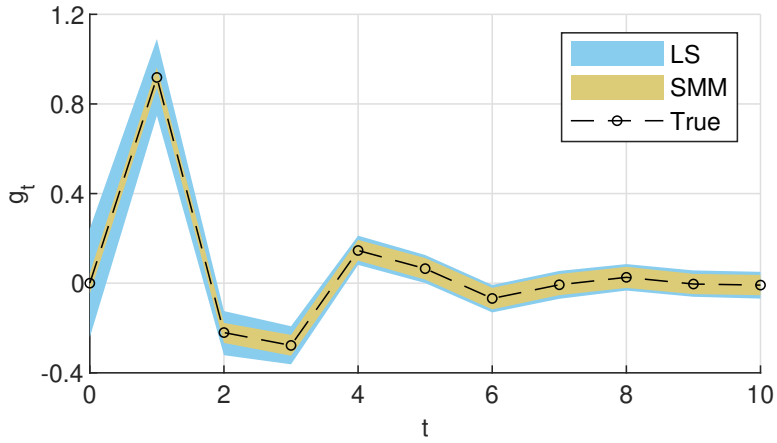



Figure 4.5: Comparison of impulse response estimation with unknown input history. Colored area: estimates within two standard deviations.

In the second example, we focus on the effect of unknown input history by investigating a faster LTI system used in Pillonetto and De Nicolao (2010):

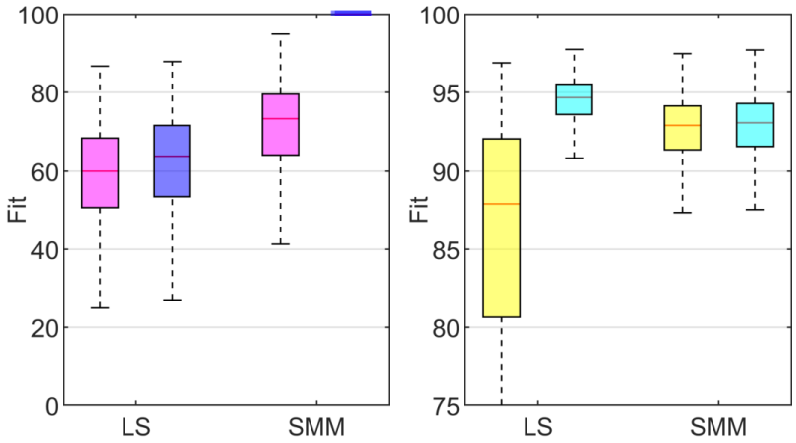
$$G_5(q) = \frac{0.9183q}{q^2 + 0.24q + 0.36}. \tag{4.67}$$

The system has been normalized to have an $\mathcal{H}_2$-norm of 1. In this case, the truncation error is already negligible at $n_g = 11$, but we assume the input history is unknown. The results of the estimation are illustrated in Figure 4.5. The results of the *SMM* algorithm are 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 $W$ in (1.8). The boxplots of model fitting for both examples are plotted in Figure 4.6. 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 significant or the input history is unknown. In example 1, the *LS* model fitting is similar for the noisy and noise-free cases, indicating that the truncation error is the primary source of error here. However, when both assumptions of the least squares are satisfied, *LS* performs slightly better than *SMM*. This is because part of the data is used to estimate the initial condition in Algorithm 4.4, whereas it is known for the *LS* algorithm.

## 4.4 Confidence Region Analysis of Prediction Errors

In this section, confidence regions are established for indirect data-driven prediction algorithms. The result first exploits information from the underlying state-space model, after which a data-driven approximation of the model information is proposed. Then, a minimum MSE predictor is proposed based on the prediction error quantification. The results are verified by numerical simulation.




(a) Example 1 (b) Example 2

Figure 4.6: Boxplots 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.

### 4.4.1 Derivation of the Confidence Region

For any stochastic data-driven predictor in the form of (4.13), the output estimate $\hat{\mathbf{y}}$ differs from the true output $\mathbf{y}$ due to the following two sources of error: 1) the output part of the signal matrix $Y_f$ is noisy, 2) the predictor estimates a trajectory whose output initial condition is $Y_p^0 g$, which differs from the trajectory to be predicted whose output initial condition is $\mathbf{y}_{\text{ini}}^0$. Consider $\Sigma_g$ defined in (4.43) and $\mathbf{v}_p \sim \mathcal{N}(\mathbf{0}, \Sigma_{\text{yini}})$. By characterizing the distributions of these two sources of error for a particular estimate of $g$ and $\delta$, we obtain the following confidence region for stochastic data-driven prediction.

**Theorem 4.2.** *Consider a data-driven predictor $\hat{\mathbf{y}} = \mathcal{F}_Z(\mathbf{u}; \mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}}) = Y_f g$ satisfying (4.13). The true output $\mathbf{y}$ is in the following ellipsoidal set with probability (w.p.) $p$:*

$$\mathcal{Y} = \left\{ \mathbf{y} \mid (\hat{\mathbf{y}} - \mathbf{y} - \Gamma\delta)^\mathsf{T} \Sigma^{-1} (\hat{\mathbf{y}} - \mathbf{y} - \Gamma\delta) \leq \mu_p \right\}, \tag{4.68}$$

*where*

$$\Gamma := \operatorname{col}\left(CA^{L_0}, \ldots, CA^{L-1}\right) \operatorname{col}\left(C, \ldots, CA^{L_0-1}\right)^\dagger, \tag{4.69}$$

$$\Sigma := \begin{bmatrix} -\Gamma & \mathbb{I}_{n_y L'} \end{bmatrix} \Sigma_g \begin{bmatrix} -\Gamma^\mathsf{T} \\ \mathbb{I}_{n_y L'} \end{bmatrix} + \Gamma \Sigma_{\text{yini}} \Gamma^\mathsf{T}, \tag{4.70}$$

*and $\mu_p$ satisfies $F_{\chi^2(n_y L')}(\mu_p) \geq p$, where $F_{\chi^2(d)}(\cdot)$ is the cumulative distribution function of the $\chi^2$-distribution with $d$ degrees of freedom.*

*Proof.* Let the stochastic noise in $Y_p$, $Y_f$, and $\mathbf{y}_{\text{ini}}$ be $E_p$, $E_f$, and $\mathbf{v}_p$, respectively, i.e.,

$$E_p := Y_p - Y_p^0,\ E_f := Y_f - Y_f^0,\ \mathbf{v}_p := \mathbf{y}_{\text{ini}} - \mathbf{y}_{\text{ini}}^0. \tag{4.71}$$

The estimation error can be decomposed as follows, according to the two aforementioned sources of error

$$\hat{\mathbf{y}} - \mathbf{y} = E_f g + \mathbf{y}^-, \tag{4.72}$$

where $\mathbf{y}^-$ is the error due to the discrepancy $\left(Y_p^0 g - \mathbf{y}_{\text{ini}}^0\right)$ in the output initial condition. The initial condition error $\mathbf{y}^-$ can be seen as the autonomous response from initial condition $\mathbf{u}_{\text{ini}}^- = \mathbf{0}$, $\mathbf{y}_{\text{ini}}^- = Y_p^0 g - \mathbf{y}_{\text{ini}}^0$. From (4.13b) and (4.71), we have

$$Y_p^0 g = \mathbf{y}_{\text{ini}} + \delta - E_p g,\ \mathbf{y}_{\text{ini}}^0 = \mathbf{y}_{\text{ini}} - \mathbf{v}_p, \tag{4.73}$$

$$\mathbf{y}_{\text{ini}}^- = (\mathbf{y}_{\text{ini}} + \delta - E_p g) - (\mathbf{y}_{\text{ini}} - \mathbf{v}_p) = \delta + \mathbf{v}_p - E_p g. \tag{4.74}$$

Let the state of the trajectory at time $-L_0$ be $x^-$. Then we have

$$\mathbf{y}_{\text{ini}}^- = \operatorname{col}\left(C, \ldots, CA^{L_0-1}\right) x^-, \quad \mathbf{y}^- = \operatorname{col}\left(CA^{L_0}, \ldots, CA^{L-1}\right) x^-. \tag{4.75}$$

Since $L_0 \geq l$, $\operatorname{col}\left(C, \ldots, CA^{L_0-1}\right)$ has full column rank. Thus, we have

$$x^- = \operatorname{col}\left(C, \ldots, CA^{L_0-1}\right)^\dagger \mathbf{y}_{\text{ini}}^-. \tag{4.76}$$

This directly leads to $\mathbf{y}^- = \Gamma \mathbf{y}_{\text{ini}}^-$. From (4.72)-(4.74), the estimation error is then

$$\hat{\mathbf{y}} - \mathbf{y} = E_f g + \Gamma(\delta + \mathbf{v}_p - E_p g). \tag{4.77}$$

Recall that $\mathbf{v}_p \sim \mathcal{N}\left(\mathbf{0}, \Sigma_{\text{yini}}\right)$, $\operatorname{col}(E_p, E_f) g \big| g \sim \mathcal{N}(\mathbf{0}, \Sigma_g)$, and they are uncorrelated. The distribution of $(\hat{\mathbf{y}} - \mathbf{y})$ given $g$ and $\delta$ is Gaussian with

$$\begin{aligned}
\mathbb{E}(\hat{\mathbf{y}} - \mathbf{y}) &= \Gamma\delta, \\
\operatorname{cov}(\hat{\mathbf{y}} - \mathbf{y}) &= \mathbb{E}\left(\begin{bmatrix} -\Gamma & \mathbb{I}_{n_y L'} \end{bmatrix} \begin{bmatrix} E_p \\ E_f \end{bmatrix} g + \Gamma \mathbf{v}_p\right)\left(\begin{bmatrix} -\Gamma & \mathbb{I}_{n_y L'} \end{bmatrix} \begin{bmatrix} E_p \\ E_f \end{bmatrix} g + \Gamma \mathbf{v}_p\right)^\mathsf{T} \\
&= \begin{bmatrix} -\Gamma & \mathbb{I}_{n_y L'} \end{bmatrix} \Sigma_g \begin{bmatrix} -\Gamma^\mathsf{T} \\ \mathbb{I}_{n_y L'} \end{bmatrix} + \Gamma \Sigma_{\text{yini}} \Gamma^\mathsf{T} = \Sigma.
\end{aligned} \tag{4.78}$$

Therefore, $(\hat{\mathbf{y}} - \mathbf{y} - \Gamma\delta)^\mathsf{T} \Sigma^{-1} (\hat{\mathbf{y}} - \mathbf{y} - \Gamma\delta)$ is subject to the $\chi^2$-distribution with $L'$ degrees of freedom. This directly leads to (4.68). □

Alternatively, the distribution of the true output trajectory can be expressed for a given predictor from (4.78).

**Corollary 4.1.** *For a given $g$, the distribution of the output trajectory $\mathbf{y}$ is given by $\mathbf{y}|g \sim \mathcal{N}(\bar{\mathbf{y}}, \Sigma)$,*

*where*

$$\bar{\mathbf{y}} := Y_f g - \Gamma(Y_p g - \mathbf{y}_{\text{ini}}). \tag{4.79}$$

**Remark 4.9.** *Theorem 4.2 still holds when the system is not observable by replacing A, C, and l with those of the observable part of the system.*

**Remark 4.10.** *The derivation is inspired by the prediction error bound presented in Section IV.C of Berberich et al. (2021). However, the results in Berberich et al. (2021) consider a bounded non-stochastic noise model and provide a deterministic but admittedly non-tight bound on* $\|\hat{\mathbf{y}} - \mathbf{y}\|$.

**Remark 4.11.** *When the noise is non-Gaussian, the statistics in Corollary 4.1 still hold, i.e.,* $\mathbb{E}[\mathbf{y}|g] = \bar{\mathbf{y}}$ *and* $\operatorname{cov}(\mathbf{y}|g) = \Sigma$.

**Remark 4.12.** *When the diagonal approximation of* $\Sigma_g$ *in Section 4.3.2 (cf. Remark 4.5) is used,* $\Sigma$ *can be simplified as*

$$\Sigma = \|g\|_2^2 T + \Gamma \Sigma_{\text{yini}} \Gamma^{\mathsf{T}}, \quad \text{where } T := \sigma^2 \left(\Gamma\Gamma^\top + \mathbb{I}\right). \tag{4.80}$$

**Remark 4.13.** *Consider the case when disturbances* $w_t$ *are present in the system (cf. Remark 4.3). Assume that the offline disturbance sequence* $\mathbf{w}^d$ *is noise-free and the online disturbance sequence is measured and predicted as* $\mathbf{w} = \mathbf{w}^0 + \varepsilon_w$*, where* $\varepsilon_w \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$*. Theorem 4.2 and Corollary 4.1 still hold with the augmented input signal matrix and the augmented input sequence by adding an additional term in* $\Sigma$ *due to the online disturbance error:*

$$\Sigma = \begin{bmatrix} -\Gamma & \mathbb{I}_{n_y L'} \end{bmatrix} \Sigma_g \begin{bmatrix} -\Gamma^{\mathsf{T}} \\ \mathbb{I}_{n_y L'} \end{bmatrix} + \Gamma \Sigma_{\text{yini}} \Gamma^\top + \Gamma_w \Sigma_w \Gamma_w^\top, \tag{4.81}$$

*where* $\Gamma_w := (Y_f - \Gamma Y_p) R_w$ *and* $R_w$ *is the last* $n_w L$ *columns of* $R_1$.

Unfortunately, the confidence region given in Theorem 4.2 is not available in practice since $\Gamma$ depends on the unknown model parameters $A$ and $C$. However, this system parameter matrix can be formulated alternatively by another data-driven prediction scheme offline. As can be seen from the proof of Theorem 4.2, the matrix $\Gamma$ can be considered as the true linear data-driven predictor $\mathbf{y} = \Gamma \mathbf{y}_{\text{ini}}$ with $\mathbf{u} = \mathbf{0}$ and $\mathbf{u}_{\text{ini}} = \mathbf{0}$. Using the certainty equivalence principle, an estimate $\hat{\Gamma}_Z$ can be found by replacing $\mathbf{y}$ with $\bar{\mathbf{y}}$. Then we have

$$\bar{\mathbf{y}} = Y_f R_2 \mathbf{y}_{\text{ini}} - \hat{\Gamma}_Z (Y_p R_2 \mathbf{y}_{\text{ini}} - \mathbf{y}_{\text{ini}}) = \hat{\Gamma}_Z \mathbf{y}_{\text{ini}}, \tag{4.82}$$

which leads to

$$\hat{\Gamma}_Z = Y_f R_2 (Y_p R_2)^{-1}. \tag{4.83}$$

This estimate is correct in the noise-free case and consistent under mild conditions, as shown in the following propositions. The validity of the estimate will be investigated numerically in Section 4.4.3.

**Proposition 4.3.** *Consider predictors in the form of* (4.14)*. If* $\sigma^2 = 0$*, Theorem 4.2 and Corollary 4.1 are still satisfied by replacing* $\Gamma$ *with* $\hat{\Gamma}_Z$*.*

*Proof.* When $\sigma^2 = 0$, all designs of $\lambda$ and $S$ are equivalent to the subspace predictor, under which case $R_2$ is the last $n_y L_0$ columns of $\mathrm{col}\,(U, Y_p)^\dagger$ and thus $Y_p R_2 = \mathbb{I}_{n_y L_0}$. According to Theorem 4.1.3, for any output initial condition $\mathbf{y}_{\mathrm{ini}}$, $\mathbf{y} = \hat{\Gamma}_Z \mathbf{y}_{\mathrm{ini}}$ is the unique autonomous response with $\mathbf{u}_{\mathrm{ini}} = \mathbf{0}$. So we have $\mathbf{y}^- = \hat{\Gamma}_Z \mathbf{y}^-_{\mathrm{ini}}$. The rest of the proof of Theorem 4.2 remains the same. □

**Remark 4.14.** *In general,* $\hat{\Gamma}_Z \neq \Gamma$*. This is because when* $n_y L_0 > n_x$*, the valid* $\Gamma$ *in the proof of Theorem 4.2 is not unique. The pseudoinverse solution (4.69) gives only one possibility.*

**Proposition 4.4.** *Consider predictors in the form of* (4.14)*. Let the singular values of* $\mathrm{col}\,(U, Y_p)$ *be* $\sigma_1, \ldots, \sigma_{L_\sigma}$ *in descending order, where* $L_\sigma := n_u L + n_y L_0$*. Then as* $M \to \infty$*, Theorem 4.2 and Corollary 4.1 hold asymptotically by replacing* $\Gamma$ *with* $\hat{\Gamma}_Z$*, if* $\sigma_{L_\sigma} \to \infty$*.*

*Proof.* Let $\mathrm{col}\,(U, Y_p) := \Omega S V^\top$ be the SVD, where $\Omega, S \in \mathbb{R}^{L_\sigma \times L_\sigma}$ and $V \in \mathbb{R}^{M \times L_\sigma}$. Then, $g_{\mathrm{pinv}} = V S^{-1} \Omega^\top \omega$, where $\omega := \mathrm{col}\,(\mathbf{u}_{\mathrm{ini}}, \mathbf{u}, \mathbf{y}_{\mathrm{ini}})$, and $\left\|g_{\mathrm{pinv}}\right\|_2^2 \leq \|V\|_2^2 \left\|S^{-1}\right\|_2^2 \|\Omega\|_2^2 \|\omega\|_2^2 = \|\omega\|_2^2 \Big/ \sigma_{L_\sigma}^2$. Note that $g_{\mathrm{pinv}}$ is also the least-norm solution to the linear system $\mathrm{col}\,(U, Y_p)\, g = \omega$, so we have $\|g\|_2^2 \leq \left\|g_{\mathrm{pinv}}\right\|_2^2$. Therefore, if $\sigma_{L_\sigma} \to \infty$, $\|g\|_2^2 \to 0$ and $\Sigma \to \mathbf{0}$ since $\Sigma_{\mathrm{yini}} = \mathbf{0}$. So we have $\mathbf{y}^- = \hat{\Gamma}_Z \mathbf{y}^-_{\mathrm{ini}}$ asymptotically. The rest of the proof of Theorem 4.2 remains the same. □

**Remark 4.15.** *The singular value condition* $\sigma_{L_\sigma} \to \infty$ *requires that the columns of* $\mathrm{col}\,(U, Y_p)$ *activate all directions persistently as* $M \to \infty$*. This is satisfied for, for example, independent random or repeated full-rank inputs.*

### 4.4.2 Minimum Mean-Squared Error Predictor

In this subsection, the distribution of the prediction error (4.78) is used to propose an optimal predictor in the form of (4.13). This algorithm finds $g$ and $\delta$ in the mapping by minimizing the expected prediction error subject to (4.78), which leads to the following proposition.

**Proposition 4.5.** *The minimum MSE estimate of the mapping in the form of (4.13) is given by*

$$\mathcal{F}_Z(\cdot) = Y_f \underset{g}{\operatorname{argmin}} \; \delta^\mathsf{T} \Gamma^\mathsf{T} \Gamma \delta + \mathrm{tr}\left( \begin{bmatrix} -\Gamma & \mathbb{I}_{n_y L'} \end{bmatrix} \Sigma_g \begin{bmatrix} -\Gamma^\mathsf{T} \\ \mathbb{I}_{n_y L'} \end{bmatrix} \right) \quad \text{s.t.} \quad (4.13\text{b}). \tag{4.84}$$

*Proof.* The MSE is calculated as

$$\begin{aligned}\mathrm{MSE}(\hat{\mathbf{y}}-\mathbf{y}) &= \mathbb{E}\left[(\hat{\mathbf{y}}-\mathbf{y})^{\mathsf{T}}(\hat{\mathbf{y}}-\mathbf{y})\right] \\ &= \mathrm{tr}\left(\mathrm{cov}(\hat{\mathbf{y}}-\mathbf{y})+\mathbb{E}(\hat{\mathbf{y}}-\mathbf{y})\,\mathbb{E}(\hat{\mathbf{y}}-\mathbf{y})^{\mathsf{T}}\right) \\ &= \mathrm{tr}\left(\Sigma+\Gamma\delta\delta^{\mathsf{T}}\Gamma^{\mathsf{T}}\right) = \mathrm{tr}(\Sigma)+\delta^{\mathsf{T}}\Gamma^{\mathsf{T}}\Gamma\delta,\end{aligned} \tag{4.85}$$

where the third equality comes from (4.78). From the definition of $\Sigma$ in (4.70), it is observed that since $\Gamma\Sigma_{\mathrm{yini}}\Gamma^{\mathsf{T}}$ does not depend on the optimization variables $g$ and $\delta$, minimizing the MSE is equivalent to the optimization problem (4.84). □

**Remark 4.16.** *If the diagonal assumption of $\Sigma_g$ is considered, (4.84) takes the unified form (4.14) with $S=\Gamma^{\mathsf{T}}\Gamma$ and $\lambda=\sigma^2 n_y L'+\sigma^2\,\mathrm{tr}(S)$.*

The implications of Proposition 4.5 are twofold. On the one hand, it provides the optimal solution to the data-driven prediction problem with output noise in terms of minimizing the MSE. Although the optimal solution relies on the unknown extended observability matrix to formulate $\Gamma$, it can be used with a preliminary model or a model set via minimax approaches.

On the other hand, similar to establishing the confidence region, the parameter $\Gamma$ used in the minimum MSE solution (4.84) can be replaced by the data-driven estimate $\hat{\Gamma}_Z$ (4.83) derived from the same signal matrix for an approximate solution. This leads to the minimum-MSE data-driven predictor, described in Algorithm 4.5.

**Algorithm 4.5** The minimum-MSE data-driven predictor with stochastic data

1: **Given:** signal matrix $Z$, noise model $\Sigma_g, \Sigma_{\mathrm{yini}}$, confidence level $p$
2: **Input:** $\mathbf{u}_{\mathrm{ini}}, \mathbf{y}_{\mathrm{ini}}, \mathbf{u}$
3: Calculate $\hat{\Gamma}_Z$ by (4.83).
4: Find $\hat{\mathbf{y}}=\mathcal{F}_Z(\mathbf{u};\mathbf{u}_{\mathrm{ini}},\mathbf{y}_{\mathrm{ini}})$ by solving (4.84) with $\Gamma=\hat{\Gamma}_Z$.
5: Find $p$-confidence region $\mathcal{Y}$ by (4.68) with $\Gamma=\hat{\Gamma}_Z$.
6: **Output:** $\hat{\mathbf{y}}$, $\mathcal{Y}$

### 4.4.3 Numerical Results

Numerical tests are conducted to illustrate the validity of the derived confidence region and the effectiveness of the proposed minimum-MSE algorithm. In the examples, stochastic data with i.i.d. noise are collected from one single experiment and used to construct $Z$ with a Page matrix construction. Unit Gaussian input sequences are used to generate the data.

First, we consider a simple two-dimensional example for illustration purposes. The prediction problem is to find the first two points ($L'=2$) in the step response of the following fourth-order system

$$G_6(q)=\frac{0.1059(0.1q^4+q^3+0.5q^2)}{q^4-2.2q^3+2.42q^2-1.87q+0.7225}. \tag{4.86}$$

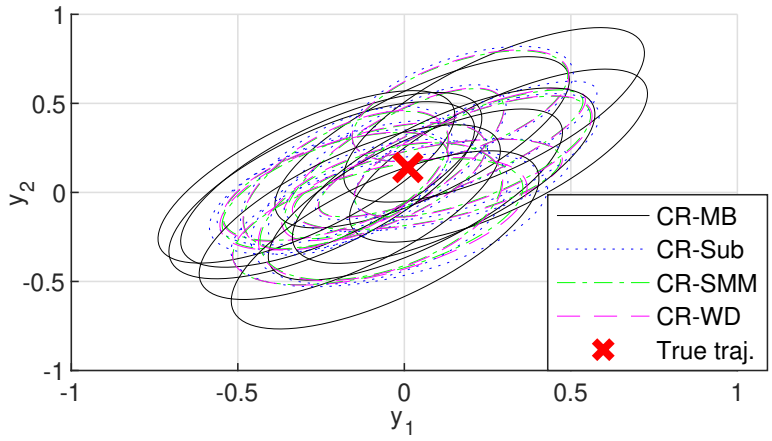



Figure 4.7: Comparison of different confidence region formulations ($p = 0.90$) tested on the *MSE-SMM* predictor with 10 different realizations of the stochastic data.

The system has been normalized to have an $\mathcal{H}_2$-norm of 1. The prediction conditions are $\mathbf{u}_{\text{ini}} = \mathbf{0}$, $\mathbf{y}_{\text{ini}} = \mathbf{0}$, and $\mathbf{u} = [1\ 1]^\mathsf{T}$. The following parameters are used: $L = 10$, $L_0 = 8$, $M = 80$, and noise level $\sigma^2 = 0.1$. A confidence level of $p = 0.90$ is used in the following figures.

Figure 4.7 compares the confidence regions obtained using model-based $\Gamma$ (4.69) (*CR-MB*) and data-driven estimates $\hat{\Gamma}_Z$, derived using the subspace predictor $\hat{\Gamma}_{\text{Sub}}$ (*CR-Sub*), the SMM predictor $\hat{\Gamma}_{\text{SMM}}$ (*CR-SMM*), and the minimum-WD predictor $\hat{\Gamma}_{\text{WD}}$ (*CR-WD*). The confidence regions are tested on the minimum-MSE predictor with data-driven $\hat{\Gamma}_{\text{SMM}}$ (*MSE-SMM*). Ten different realizations of the stochastic data are plotted. The results show that the data-driven formulations (*CR-Sub*, *CR-SMM*, and *CR-WD*) obtain similar confidence regions but are different from the model-based formulation. This is because the data-driven formulations estimate the noise-free $\hat{\Gamma}_Z$ that differs from the model-based $\Gamma$. Nevertheless, all the confidence regions are valid for this problem since the true trajectory lies in the regions with high probability.

Then, the sizes of the confidence regions are analyzed for different stochastic data-driven predictors. The following predictors are compared: 1) subspace predictor (4.19) (*Sub*), 2) signal matrix model (Algorithm 4.4) (*SMM*), 3) minimum-WD predictor (4.22) (*WD*), and 4) minimum-MSE predictor using model-based $\Gamma$ (4.69) (*MSE-MB*), data-driven $\hat{\Gamma}_{\text{Sub}}$ (*MSE-Sub*), $\hat{\Gamma}_{\text{SMM}}$ (*MSE-SMM*), and $\hat{\Gamma}_{\text{WD}}$ (*MSE-WD*). Figure 4.8 shows the confidence regions of these stochastic predictors with model-based $\Gamma$ (*CR-MB*). As can be seen from the figure, the existing algorithms (*Sub*, *SMM*, and *WD*) have larger confidence regions compared to the minimum-MSE algorithms (*MSE-MB* and *MSE-SMM*). This illustrates the effectiveness of the proposed algorithm in improving prediction accuracy. In this example, the confidence regions of *MSE-Sub* and *MSE-WD* are very close to that of *MSE-SMM*, so they are omitted in Figure 4.8.

To quantitatively assess the derived confidence region and the minimum-MSE prediction algorithm, the following campaign of 1000 Monte Carlo simulations is set up. A bank of 1000 SISO systems is randomly generated by the `drss` command in MATLAB with random numbers of

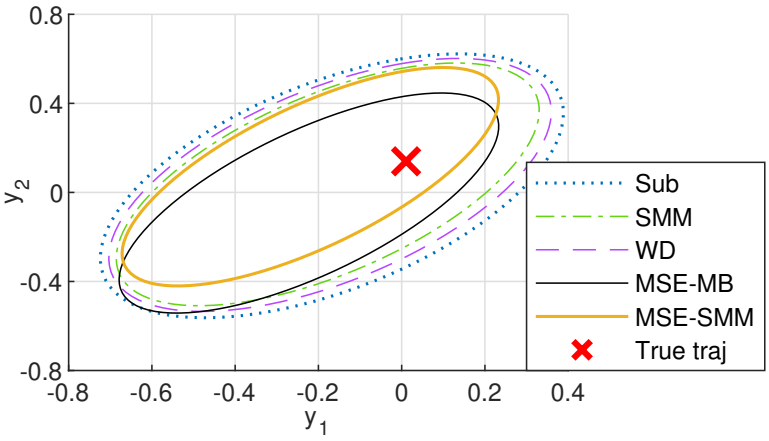



Figure 4.8: Comparison of different stochastic data-driven predictors in terms of the confidence regions ($p = 0.90$) with model-based $\Gamma$ (*CR-MB*).

Table 4.1: Empirical confidence levels of the confidence regions.

| $p = 0.95$ | CR-MB | CR-Sub | CR-SMM | CR-WD |
|---|---|---|---|---|
| Sub | 97.1% | 98.7% | 98.4% | 98.7% |
| SMM | 96.8% | 97.4% | 97.3% | 97.3% |
| MSE-SMM | 95.2% | 96.4% | 96.2% | 96.4% |
| $p = 0.99$ | CR-MB | CR-Sub | CR-SMM | CR-WD |
| Sub | 99.3% | 100% | 99.8% | 99.9% |
| SMM | 99.2% | 99.7% | 99.7% | 99.7% |
| MSE-SMM | 99.0% | 99.3% | 99.2% | 99.3% |

states between 3 and 8. These random systems are normalized to have an $\mathcal{H}_2$-gain of 1. The prediction problem uses the following parameters: $L = 20$, $L_0 = 8$, $L' = 12$, and $M = 320$. The input $\mathbf{u}$ and the initial condition $(\mathbf{u}_{\text{ini}}, \mathbf{y}_{\text{ini}})$ are selected randomly with a unit Gaussian distribution.

Table 4.1 compares the percentage of the simulations where the true response is in the confidence region, i.e., $\mathbf{y}^i \in \mathcal{Y}^i$ for the $i$-th simulation, for the model-based and different data-driven formulations. Two confidence levels, $p = 0.95$ and $p = 0.99$, are selected. The noise level is selected as $\sigma^2 = 0.1$. The rows in Table 4.1 correspond to different predictors, whereas the columns correspond to different formulations of the confidence region. It can be seen from the table that the empirical confidence levels match the targeted $p$-value well with the model-based $\Gamma$ (*CR-MB*) for all three predictors, where Theorem 4.2 is satisfied exactly. With the data-driven estimates $\hat{\Gamma}_Z$, the confidence regions become marginally more conservative as the empirical confidence levels are slightly larger in Table 4.1. The results of the three data-driven estimates (*CR-Sub*, *CR-SMM*, *CR-WD*) are similar, which indicates that the confidence region is not very sensitive to the choice of the $\hat{\Gamma}_Z$ estimation method.

Table 4.2 compares the empirical MSE of the predictors in the Monte Carlo simulations to the

Table 4.2: Comparison of the estimated and the empirical MSE.

| $\sigma^2 = 0.1$ | Empirical | CR-Sub | CR-SMM | CR-WD |
|---|---|---|---|---|
| Sub | 0.115 | 0.153 | 0.149 | 0.152 |
| SMM | 0.099 | 0.142 | 0.137 | 0.140 |
| MSE-SMM | 0.096 | 0.136 | 0.131 | 0.134 |
| $\sigma^2 = 1$ | Empirical | CR-Sub | CR-SMM | CR-WD |
| Sub | 1.106 | 1.529 | 1.485 | 1.511 |
| SMM | 0.915 | 1.391 | 1.344 | 1.372 |
| MSE-SMM | 0.897 | 1.335 | 1.286 | 1.317 |

Table 4.3: Comparison of the empirical MSE for different predictors.

| | $\sigma^2 = 0.1$ | $\sigma^2 = 0.5$ | $\sigma^2 = 1$ |
|---|---|---|---|
| Sub | 0.115 | 0.558 | 1.106 |
| SMM | 0.099 | 0.476 | 0.915 |
| WD | 0.113 | 0.548 | 1.091 |
| MSE-MB | 0.094 | 0.435 | 0.833 |
| MSE-Sub | 0.097 | 0.464 | 0.908 |
| MSE-SMM | 0.096 | 0.460 | 0.897 |
| MSE-WD | 0.097 | 0.462 | 0.902 |

MSE estimated by (4.85) with the approximate data-driven confidence regions. The empirical MSE is computed as $\mathrm{MSE}_{\mathrm{emp}}(\hat{\mathbf{y}} - \mathbf{y}) := \frac{1}{N_s}\sum_{i=1}^{N_s} \left\|\hat{\mathbf{y}}^i - \mathbf{y}^i\right\|_2^2$, where $\hat{\mathbf{y}}^i$ and $\mathbf{y}^i$ are the predicted and the true responses of the $i$-th simulation, respectively, and $N_s = 1000$. Two different noise levels of $\sigma^2 = 0.1$ and $\sigma^2 = 1$ are considered. Similar to the observation from Table 4.1, the estimated MSE is more conservative than the empirical ones for all three predictors. It is also observed that the region *CR-SMM* is the less conservative among those tested here. Nevertheless, the estimated MSE can correctly predict the relative error magnitudes of different predictors. This illustrates that the estimated MSE can be a good indicator of prediction accuracy. Only three representative predictors are shown in Table 4.1 and Table 4.2 for clarity. The results of the other algorithms are similar.

Finally, we compare the prediction accuracy of the predictors by the empirical MSE under three different noise levels: $\sigma^2 = 0.1$, $\sigma^2 = 0.5$, and $\sigma^2 = 1$. The results are shown in Table 4.3. For all three noise levels, the minimum-MSE predictor with model-based $\Gamma$ (*MSE-MB*) achieves the minimum empirical MSE. This is expected as *MSE-MB* exactly optimizes for this objective as demonstrated in Proposition 4.5. However, the model-based $\Gamma$ is not available in practice. Among the other practical algorithms, *MSE-SMM* has the smallest empirical MSE, with slightly better performance than the direct SMM approach (*SMM*). This result shows numerically that, with approximate data-driven formulations of $\hat{\Gamma}_Z$, the proposed minimum-MSE predictor still obtains a more accurate prediction than the existing algorithms.

## 4.5 Summary

This chapter investigates the problem of nonparametric data-driven trajectory prediction with stochastic data. Based on the Willems' fundamental lemma for deterministic data, the stochastic problem can be tackled in two directions.

In the first direction, the stochastic data are denoised by solving a low-rank Hankel matrix denoising problem. Instead of finding a low-rank approximation of the noisy matrix, the proposed approach applies the singular value shrinkage law that is asymptotically optimal in terms of estimating the noise-free matrix. Together with an iterative algorithm to enforce the generalized Hankel structure, this algorithm achieves the best noise reduction performance numerically compared to other low-rank approximation or denoising algorithms.

In the second direction, a novel statistical framework, dubbed the signal matrix model, is proposed to obtain a tuning-free stochastic data-driven predictor based on maximum likelihood estimation. The problem is solved efficiently by approximating it as a sequential quadratic program with data compression. The proposed predictor performs better than the subspace predictor and can obtain impulse response estimates with less restrictive assumptions than the least-squares method.

Finally, the prediction error of data-driven predictors with stochastic data is characterized statistically. The framework provides ellipsoidal confidence regions for various predictors. It also offers an optimal predictor that minimizes the mean-squared prediction error directly. In practice, both the confidence region and the minimum mean squared error predictor can be implemented with data-driven approximations that show good accuracy numerically.

# 5 Predictive Control with Data-Driven Predictors

This chapter focuses on designing predictive controllers with the nonparametric data-driven predictors studied in Chapter 4. This approach, known as data-driven predictive control (DDPC), is the data-driven counterpart to model predictive control (MPC), which solves a finite-horizon optimal control problem in a receding horizon fashion. Compared with model-based predictors, which typically provide one-step-ahead prediction at one time, data-driven trajectory predictors are more convenient by providing multi-step-ahead prediction directly.

With deterministic data, trajectory characterization from the Willems' fundamental lemma (WFL) can be used as an implicit predictor. This idea has attracted significant attention since its proposal in Coulson et al. (2019) under the name of data-enabled predictive control (DeePC). As discussed in Section 4.1.2, this characterization is ill-defined with uncertainties. This issue can be remedied by adding regularization terms to the control cost to penalize unreliable predictions. This idea, known as direct DDPC or regularized DeePC, has been extensively studied under the robust control framework with a bounded uncertainty set, including distributionally robust optimization (Coulson et al., 2022), performance guarantee (Berberich et al., 2021; Huang et al., 2023), and constraint tightening (Berberich et al., 2020; Klöppelt et al., 2022). Still, it is unclear how the regularization parameters should be designed in practice and how the control actions can be interpreted under ill-defined predictions. On the other hand, research under the stochastic control framework is limited, where existing works adopt restrictive assumptions, such as noise-free offline data (Kerz et al., 2023) and exact polynomial chaos expansions of stochastic measurements (Pan et al., 2023).

Section 5.1 investigates the indirect DDPC algorithm by adopting the well-defined indirect data-driven predictor discussed in Sections 4.3, namely the signal matrix model (SMM), with certainty equivalence. This algorithm is named signal matrix model predictive control (SMM-PC). The control performance of the proposed algorithm is shown to be better than the pseudoinverse subspace predictor and the direct DDPC algorithm with optimal regularization parameters by oracle, especially under low SNRs. Online data can also be incorporated into the signal matrix to make the algorithm adaptive, improving system knowledge online and extending the algorithm to

slowly time-varying systems.

However, the certainty-equivalent implementation suffers from the following problems: 1) the control cost does not account for the prediction error, 2) the initial condition measurements suffer from noise, which cannot be improved by collecting more data, and 3) the constraint satisfaction is not guaranteed. Section 5.2 addresses these problems under general unbounded stochastic uncertainties, utilizing the prediction error quantification provided in Section 4.4. In particular, three modifications are made to the certainty-equivalent DDPC algorithm. 1) The nominal control cost is replaced by the expected control cost. This introduces an additional uncertainty term resembling the regularizer used in regularized DeePC, but the weight is statistically specified without tuning. 2) The output initial condition is estimated by Kalman-filtering the output measurements with one-step-ahead predictions from the previous time instant. This significantly reduces the prediction error. 3) Output constraints are formulated as chance constraints and guaranteed by second-order cone (SOC) constraints. The effectiveness of these modifications is tested in a numerical example, where satisfactory control performance in terms of both control cost and constraint satisfaction is observed with significantly improved initial condition estimation.

This chapter concludes by applying the proposed algorithm to a space heating control case study in high-fidelity simulation. The control performance is extensively analyzed by comparing it against predictive control algorithms using subspace identification and direct & indirect DDPC. Results demonstrate that the proposed stochastic SMM-PC algorithm satisfies operating constraints more reliably than competing methods and reduces energy consumption simultaneously.

## 5.1 Data-Driven Predictive Control with Signal Matrix Model

We are interested in designing a receding horizon control algorithm with data-driven predictors derived from the signal matrix $Z$ instead of the state-space model (1.2). These algorithms are known as data-driven predictive control (DDPC). In particular, consider the stochastic optimal tracking problem within a horizon of $L'$ that minimizes the following expected control cost

$$\min_{(u_k^t)_{k=0}^{L'-1}} J_{\text{ctr}} := \sum_{k=0}^{L'-1} \left\|u_k^t\right\|_R^2 + \mathbb{E}\left[\sum_{k=0}^{L'-1} \left\|y_k^t - r_{t+k}\right\|_Q^2\right] \tag{5.1}$$

at time $t$ (Kouvaritakis and Cannon, 2016), where $u_k^t$ is the designed input at time $(t+k)$, $y_k^t$ is a random variable that predicts the noise-free future output $y_{t+k,0}$, $r_t$ denotes the reference trajectory, and $R, Q$ are the input and the output cost matrices, respectively. The superscript $t$ is used to denote variables at time $t$. It is also desired to constrain the outputs within a polytopic set $\mathscr{Y}_t := \{y_t \,|\, H^t y_t \leq q^t\}$ at time $t$, where $H^t := \begin{bmatrix} h_1^t & \dots & h_{n_c}^t \end{bmatrix}^\top \in \mathbb{R}^{n_c \times n_y}$ and $q^t := \operatorname{col}\left(q_1^t, \dots, q_{n_c}^t\right) \in \mathbb{R}^{n_c}$. Similarly, the inputs are constrained to be in the set $\mathscr{U}^t$ at time $t$. These lead to the constraints $u_k^t \in \mathscr{U}_{t+k}$, $y_k^t \in \mathscr{Y}_{t+k}$, $\forall k = 0, \dots, L'-1$.

The optimization problem is solved under the constraint that $\mathbf{u}^t := (u_k^t)_{k=0}^{L'-1}$ and $\mathbf{y}^t := (y_k^t)_{k=0}^{L'-1}$ are a possible trajectory of the system (1.2) under the initial condition at time $t$. In conventional MPC algorithms, this constraint is enforced by iteratively propagating the state-space model (1.2) subject to the initial state $x_t$, assuming the knowledge of the model parameters is available. In DDPC, according to Theorem 4.1.3, this constraint can be characterized by (4.4) subject to the input and output initial condition of length $L_0$, i.e., $\mathbf{u}_{\text{ini}}^t := (u_k)_{k=t-L_0}^{t-1}$ and $\mathbf{y}_{\text{ini}}^t := (y_k)_{k=t-L_0}^{t-1}$, assuming deterministic data without noise is available.

At each time instant, the first entry in the newly optimized input trajectory is applied to the system in a receding horizon fashion, i.e., $u_t := u_0^t$, and the noisy output $y_t = y_{t,0} + v_t$ is measured.

In this section, we take the certainty equivalence approach and assume that the estimated output trajectory $\hat{\mathbf{y}}$ is the same as the true trajectory $\mathbf{y}$. Modifications taking the stochasticity of the prediction into account are discussed in Section 5.2.

### 5.1.1 Data-Enabled Predictive Control

When the system is deterministic without uncertainties and the rank condition (4.3) on data informativity is satisfied, the following DDPC or unregularized DeePC algorithm based on the output predictions via (4.4) is equivalent to MPC with exact model knowledge:

$$\begin{aligned} \min_{\mathbf{u}^t, \mathbf{y}^t, g^t} \quad & J_{\text{ctr}}(\mathbf{u}^t, \mathbf{y}^t) && \text{(5.2a)} \\ \text{s.t.} \quad & \text{col}\left(\mathbf{u}_{\text{ini}}^t, \mathbf{u}^t, \mathbf{y}_{\text{ini}}^t, \mathbf{y}^t\right) = Z g^t, && \text{(5.2b)} \\ & u_k^t \in \mathscr{U}_{t+k},\ y_k^t \in \mathscr{Y}_{t+k},\ \forall k = 0, \dots, L'-1. && \text{(5.2c)} \end{aligned}$$

However, when uncertainties are present, the condition (5.2b) no longer serves as a well-defined predictor to predict $\mathbf{y}^t$ (cf. Section 4.1.2). If we still want to use this implicit characterization in DDPC, additional regularization is required in the control cost to penalize unreliable predictions (Coulson et al., 2019; Dörfler et al., 2023). This becomes the direct regularized DeePC algorithm in the form of

$$\begin{aligned} \min_{\mathbf{u}^t, \mathbf{y}^t, g^t} \quad & J_{\text{ctr}}(\mathbf{u}^t, \mathbf{y}^t) + \lambda_g \left\| \Pi g^t \right\|_p^p + \lambda_y \left\| Y_p g^t - \mathbf{y}_{\text{ini}}^t \right\|_2^2 && \text{(5.3a)} \\ \text{s.t.} \quad & \text{col}\left(\mathbf{u}_{\text{ini}}^t, \mathbf{u}^t\right) = U g^t,\ \mathbf{y}^t = Y_f g^t,\ \text{(5.2c)}, && \text{(5.3b)} \end{aligned}$$

where $\lambda_g$, $\lambda_y$ are design parameters, $\Pi$ can be selected as $\mathbb{I}$ or $\mathbb{I} - \text{col}\,(U, Y_p)^\dagger \text{col}\,(U, Y_p)$, and $p$ is either 1 or 2. The two regularization terms resemble the cost function in the indirect data-driven predictor (4.14). So, the regularized control cost can be seen as a trade-off between the control performance objective $J_{\text{ctr}}$ and the trajectory prediction objective. Unlike conventional predictive control, 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. In addition, no systematic approaches have been proposed to tune $\lambda_g$ and $\lambda_y$, but the control performance is

known to be very sensitive to the regularization parameters (Huang et al., 2019). Exhaustive parameter tuning can be extremely time-intensive for systems with large time constants.

### 5.1.2 Indirect Bi-Level Data-Driven Predictive Control

To avoid the interpretability and the parameter tuning issues discussed above, the rest of this chapter focuses on another popular method to apply DDPC with uncertainties by replacing the ill-defined implicit trajectory characterization (5.2b) with the explicit optimization-based predictor (4.14). This is known as indirect DDPC. By introducing another optimization problem as a constraint, the optimal control problem then becomes a bi-level problem:

$$\begin{aligned}
\min_{\mathbf{u}^t,\mathbf{y}^t,g^t} \quad & J_{\text{ctr}}(\mathbf{u}^t,\mathbf{y}^t) && \text{(5.4a)}\\
\text{s.t.} \quad & \mathbf{y}^t = Y_f g^t,\ u_k^t \in \mathscr{U}_{t+k},\ y_k^t \in \mathscr{Y}_{t+k},\ \forall k = 0,\dots,L'-1, && \text{(5.4b)}\\
& g^t = \underset{g}{\operatorname{argmin}}\ \left\|Y_p g - \mathbf{y}_{\text{ini}}^t\right\|_S^2 + \lambda \left\|g\right\|_2^2 \\
& \qquad \text{s.t.} \quad \operatorname{col}\left(\mathbf{u}_{\text{ini}}^t, \mathbf{u}^t\right) = Ug. && \text{(5.4c)}
\end{aligned}$$

Although it is generally difficult to solve bi-level optimization problems, as discussed in Section 4.1.2, the inner problem (5.4c) admits a linear closed-form solution:

$$g^t = R_1 \operatorname{col}\left(\mathbf{u}_{\text{ini}}^t, \mathbf{u}^t\right) + R_2 \mathbf{y}_{\text{ini}}^t. \tag{5.5}$$

So (5.4c) can be implemented as linear equality constraints, and thus (5.4) becomes tractable.

**Remark 5.1.** *The data compression scheme discussed in Section 4.3.3 applies to both the regularized DeePC and the indirect bi-level DDPC algorithms.*

One notable special case of the indirect DDPC algorithm (5.4) is the subspace predictive control (SPC) approach (Favoreel et al., 1999; Sedghizadeh and Beheshti, 2018) by adopting the subspace predictor with $S = \mathbb{I}$ and $\lambda \to 0^+$, i.e., $g^t = g_{\text{pinv}}^t := \operatorname{col}(U_p, U_f, Y_p)^\dagger \operatorname{col}(\mathbf{u}_{\text{ini}}^t, \mathbf{u}^t, \mathbf{y}_{\text{ini}}^t)$.

As Section 4.3 illustrates, the signal matrix model (SMM) provides more accurate predictions than the subspace predictor. Therefore, it is desired to use SMM (Algorithm 4.4) as the inner predictor. However, unlike the QP (5.4c) with a linear closed-form solution, the SQP iterative algorithm (4.60) does not have a closed-form solution. This makes the optimization problem computationally challenging, especially in an online scheme.

To address this problem, we notice numerically that the $l_2$-norm of $g^t$ does not change much throughout the receding horizon control, and the algorithm is only iterative with respect to $\|g\|_2^2$. So, it makes sense to warm-start the optimization problem from $g^{t-1}$ at the previous time instant. Then, the SMM predictor can be approximated by the solution of (4.60) after the first iteration, i.e.,

$$g^t = R_1(g^{t-1}) \operatorname{col}\left(\mathbf{u}_{\text{ini}}^t, \mathbf{u}^t\right) + R_2(g^{t-1})\, \mathbf{y}_{\text{ini}}^t. \tag{5.6}$$

Table 5.1: Summary of indirect DDPC designs.

| | SPC | Min-WD | SMM-PC | Min-MSE |
|---|---|---|---|---|
| $S$ | $\mathbb{I}$ | $\mathbb{I}$ | $\mathbb{I}$ | $\hat{\Gamma}_Z^{\mathsf{T}}\hat{\Gamma}_Z$ |
| $\lambda$ | $0^+$ | $n_y L_0 \sigma^2$ | $L'\sigma_p^2 \Big/ \left\|g^{t-1}\right\|_2^2 + L\sigma^2$ | $\sigma^2 n_y L' + \sigma^2 \operatorname{tr}(S)$ |

This corresponds to selecting $S = \mathbb{I}$ and $\lambda = L'\sigma_p^2 \Big/ \left\|g^{t-1}\right\|_2^2 + L\sigma^2$ in (5.4c) and is referred to as the linearized SMM predictor in what follows. This approach is named signal matrix model predictive control (SMM-PC).

**Remark 5.2.** *An alternative approach is to initialize the problem from the pseudoinverse solution corresponding to the ideal trajectory* $\mathbf{u}_{\text{ideal}}^t = \mathbf{0}$, $\mathbf{y}_{\text{ideal}}^t = (r_k)_{k=t}^{t+L'-1}$, *i.e.,* $\lambda = L'\sigma_p^2 \Big/ \left\|g_{\text{ini}}^t\right\|_2^2 + L\sigma^2$, *where* $g_{\text{ini}}^t := Z^\dagger \operatorname{col}\left(\mathbf{u}_{\text{ini}}^t, \mathbf{u}_{\text{ideal}}^t, \mathbf{y}_{\text{ini}}^t, \mathbf{y}_{\text{ideal}}^t\right)$.

Similarly, the minimum-WD and the minimum-MSE predictors can also be used in implementing indirect DDPC (5.4). Different designs of $S$ and $\lambda$ in (5.4c) are summarized in Table 5.1.

Compared with the regularized DeePC algorithm (5.3), the indirect DDPC algorithm (5.4) adopts an explicit input-output mapping, and the design parameters are specified from the data-driven predictor design without additional tuning.

### 5.1.3 Performance of Signal Matrix Model Predictive Control

This subsection analyzes the performance of SMM-PC by numerical examples tested on the fourth-order LTI system (3.26). The following parameters are used in the simulation: $L_0 = n_x = 4$, $L' = 11$, and $Q = R = 1$. Assuming the same sensor for offline and online measurements, we select $\sigma^2 = \sigma_p^2$. No input and output constraints are enforced, i.e., $\mathscr{U} = \mathscr{Y} = \mathbb{R}^{L'}$. The noise level $\sigma^2 = \sigma_p^2$ is estimated as described in Remark 4.4 for implementing SMM-PC. The offline input-output trajectory data are collected with unit i.i.d. Gaussian input signals. The control performance is assessed by the true total control cost $J_{\text{tot}}$ (1.10) over all time instants.

First, to assess the validity of the linearized SMM predictor, we investigate the discrepancy between the linearized and the iterative nonlinear SMM in the following example.

**Example 5.1.** *Consider input-output trajectory data of length* $N = 50$ *with noise level* $\sigma^2 = \sigma_p^2 = 0.1$. *The control objective is to track a sinusoidal reference trajectory* $r_t = 0.5\sin\left(\frac{\pi}{10}t\right)$ *of length* $N_c = 120$. *After implementing SMM-PC with the linearized SMM predictor, the linearized SMM predictions, denoted by* $\hat{\mathbf{y}}_{\text{LSMM}}$, *are compared to Algorithm 4.4 with the newly optimized input sequence, denoted by* $\hat{\mathbf{y}}_{\text{SMM}}$, *at each time instant. Their discrepancies are quantified by*

$$\mathrm{E} := \frac{\left\|\hat{\mathbf{y}}_{\text{LSMM}} - \hat{\mathbf{y}}_{\text{SMM}}\right\|_2}{\left\|\hat{\mathbf{y}}_{\text{SMM}}\right\|_2}. \tag{5.7}$$

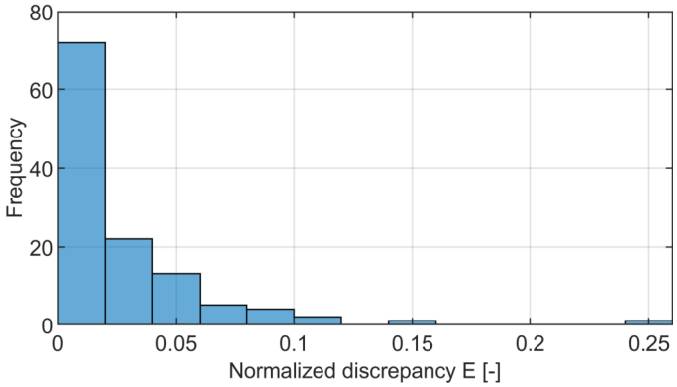



Figure 5.1: Normalized discrepancy between the linearized SMM and the iterative SMM.

*The histogram of E is plotted in Figure 5.1.*

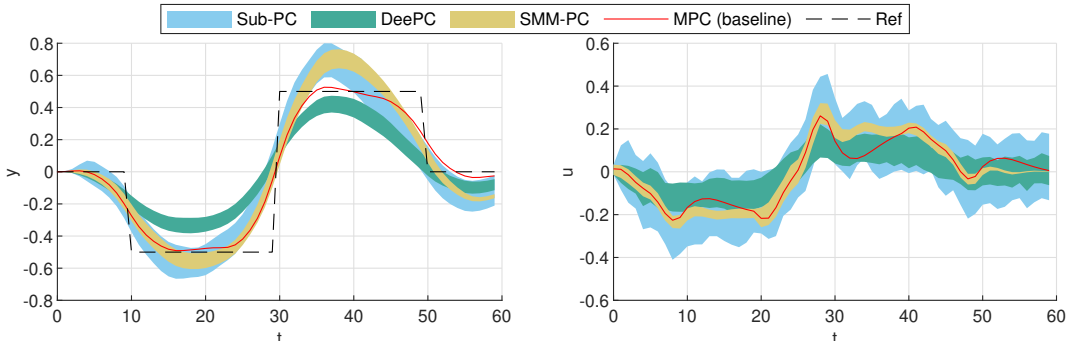
It can be seen from Figure 5.1 that the error resulting from the linearization of SMM is less than 5% for most time instants, which shows that the linearized SMM can still obtain an accurate input-output mapping.

Then, the control performances of three DDPC algorithms are compared: 1) *Sub-PC*: subspace predictive control, 2) *DeePC*: regularized data-enabled predictive control (5.3), and 3) *SMM-PC*: signal matrix model predictive control. In *DeePC*, the algorithm is tested on a nine-point logarithmic grid of $\lambda_g$ between 10 and 1000. In this example, the control performance is found to be insensitive to the value of $\lambda_y$, so a fixed value of $\lambda_y = 1000$ is used. To benchmark the performance, the ideal MPC algorithm (denoted by *MPC*) is also considered, 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.

**Example 5.2.** *Consider input-output trajectory data of length $N = 200$ with noise level $\sigma^2 = \sigma_p^2 = 1$. A square-wave reference trajectory of length $N_c = 60$, labeled Ref in Figure 5.2(a), is to be tracked. For each case, 100 Monte Carlo simulations are conducted. The closed-loop input-output trajectories of different control algorithms are plotted in Figure 5.2. These trajectories are characterized by the range within one standard deviation of the Monte-Carlo simulation. The boxplot of the control performance measure $J_{\text{tot}}$ is shown in Figure 5.3.*

The *SMM-PC* algorithm obtains the closest match to the benchmark trajectory *MPC*. *Sub-PC* applies more aggressive control inputs with much higher input costs, whereas the control strategy of *DeePC* is more conservative with more significant tracking errors. *SMM-PC* also has the smallest variance of input trajectories against different noise realizations. Figure 5.3 confirms that *SMM-PC* performs better than *Sub-PC* and *DeePC* in this control task.

When comparing the control performance, the best choices of $\lambda_g$ in *DeePC* are selected with an oracle for each run as plotted in Figure 5.4 (green) for different noise levels. It can be seen




(a) Output trajectory (b) Input trajectory

Figure 5.2: Comparison of closed-loop input-output trajectories with different control algorithms ($\sigma^2 = \sigma_p^2 = 1, N = 200$). Colored area: trajectories within one standard deviation.

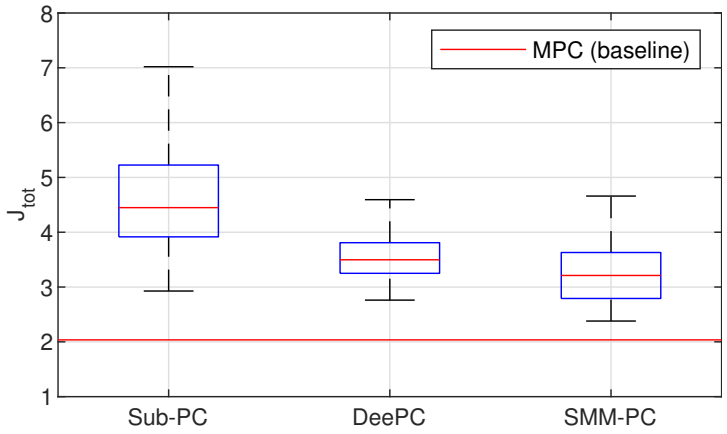



Figure 5.3: Boxplot of the control performance in terms of the true total control cost $J_{\text{tot}}$ with different control algorithms ($\sigma^2 = \sigma_p^2 = 1, N = 200$).

that, even for the same control task, the optimal value of this parameter 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. Nevertheless, the optimal value of $\lambda_g$ is used in all simulations despite being unrealistic in practice. On the other hand, the noise level estimator (4.36) used in *SMM-PC* is very effective in estimating $\sigma^2$ as demonstrated in Figure 5.4 (yellow).

The optimization problems are formulated as QP and solved by MOSEK. The computation time for the three DDPC algorithms is similar. The effect of the proposed data compression scheme in Section 4.3.3 is illustrated in Figure 5.5 with the example of *SMM-PC*. By applying the preconditioning, the online computational complexity no longer depends on the data size $N$.

Finally, the sensitivity to different offline data sizes $N$ and noise levels $\sigma^2$ is investigated in Figure 5.6. As shown in Figure 5.6(a), the control performance of *SMM-PC* is not very sensitive to the number of data points and performs uniformly better among the three algorithms. Good performance has already been obtained with only $N = 75$ points. *DeePC* does not perform very

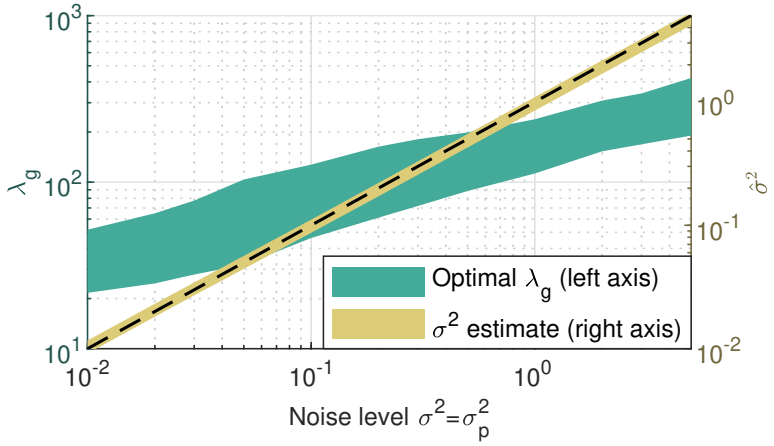



Figure 5.4: Parameter tuning in *DeePC* ($\lambda_g$) and *SMM-PC* ($\sigma^2$) for different noise levels. Colored area: values within one standard deviation. The dashed line shows the true noise level.

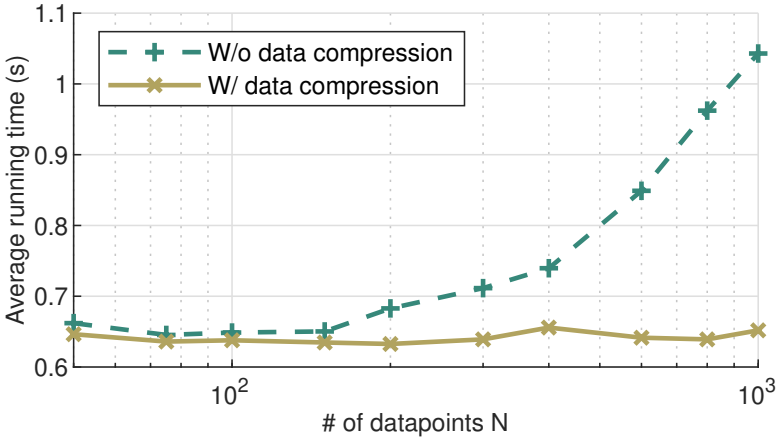
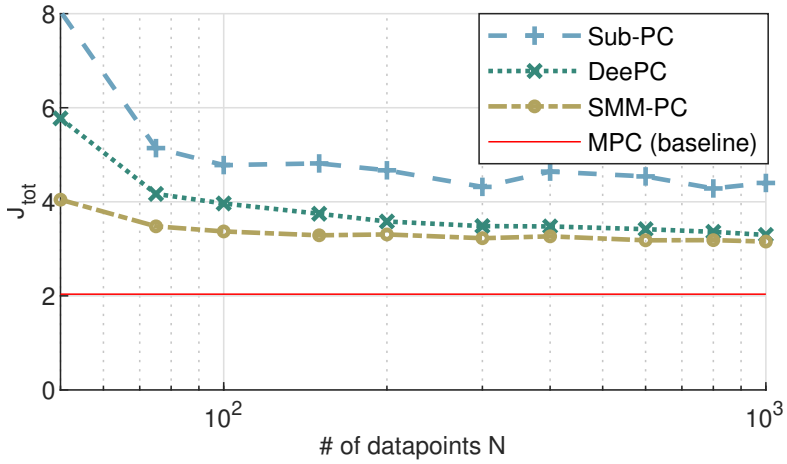



Figure 5.5: Average computation time of *SMM-PC* with and without data compression.

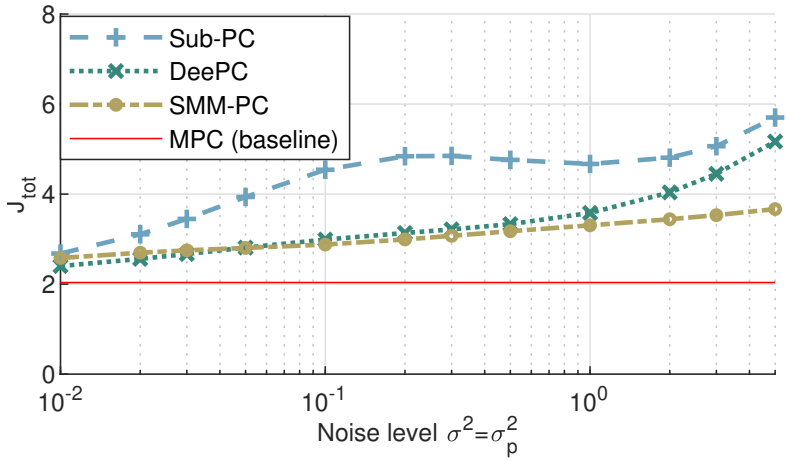
well with small data sizes but obtains a similar performance to *SMM-PC* for large $N$. *Sub-PC* cannot achieve a satisfying result even with a large data size because, as discussed in Section 4.3.4, the subspace predictor is problematic to deal with online measurement noise $\sigma_p^2$, which cannot be averaged out by a large $N$. Figure 5.6(b) shows that all three algorithms perform similarly at low noise levels as they are all stochastic variants of the noise-free algorithm (5.2). *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 advantage of *SMM-PC* is significant for higher noise levels.

### 5.1.4 Incorporation of Online Data

In addition to the offline data $(\mathbf{u}^d, \mathbf{y}^d)$, online measurements can also be used to construct the signal matrix. This improves the knowledge of the unknown system as the predictive controller is




(a) Performance as a function of the number of data points




(b) Performance as a function of the noise level

Figure 5.6: The effect of different offline data sizes and noise levels on the control performance.

deployed. This is particularly useful when the offline data quality is poor or the model parameters vary online. This strategy can be interpreted as an equivalent of adaptive MPC (Fukushima et al., 2007) for the SMM.

At each time instant $t \geq L$, a new column can be added to the signal matrix with

$$\begin{bmatrix} U_{t+1} \\ Y_{t+1} \end{bmatrix} := \begin{bmatrix} \gamma U_t & (u_k)_{k=t-L+1}^{t} \\ \gamma Y_t & (y_k)_{k=t-L+1}^{t} \end{bmatrix}, \tag{5.8}$$

where $Z_t := \mathrm{col}\,(U_t, Y_t)$ is the adaptive signal matrix applied to the SMM-PC algorithm at time $t$, and $\gamma$ is the forgetting factor of previous trajectories. The factor $\gamma$ can be chosen as 1 for LTI systems and $\gamma \in (0,1)$ when model variations are expected. Like the offline signal matrix, the adaptive signal matrix can also be compressed online to maintain $2L$ columns. The incremental SVD algorithm in Brand (2002) can be applied to reduce the computational complexity of

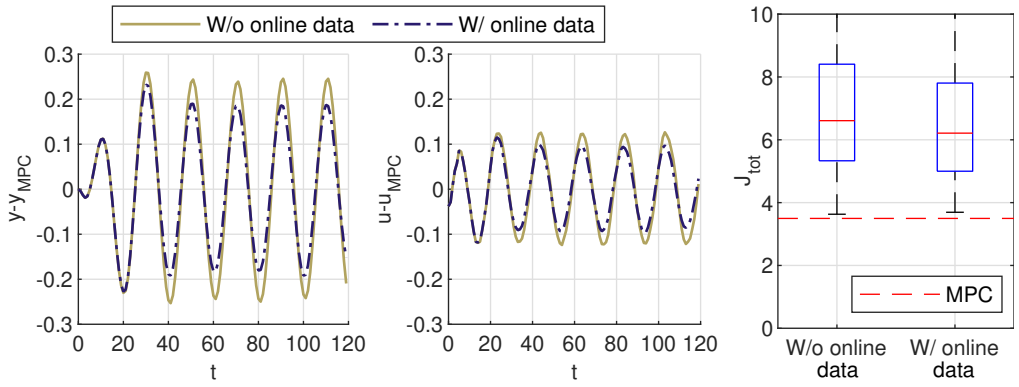



Figure 5.7: Effects of online data adaptation in SMM-PC for datasets with high noise levels. Left: deviation from ideal MPC, right: boxplot of true total control cost $J_{\text{tot}}$.

updating the compressed signal matrix, which makes use of the fact that the SVD of $\text{col}(U_t, Y_t)$ in (5.8) is already known from the previous time instant.

The following two examples investigate the effects of incorporating online data under high noise levels and with slowly time-varying parameters, respectively.

**Example 5.3.** *Consider input-output trajectory data of length $N = 100$ with a high noise level of $\sigma^2 = \sigma_p^2 = 1$. The same control objective as Example 5.1 is used. The SMM-PC algorithms are compared with the fixed signal matrix $\text{col}(U, Y)$ and the adaptive signal matrix $\text{col}(U_t, Y_t)$ with $\gamma = 1$. The deviations of the closed-loop trajectories from the ideal MPC trajectory are plotted on the left of Figure 5.7. In addition, 200 Monte Carlo simulations are conducted, and the boxplot of the true total control cost $J_{\text{tot}}$ is shown on the right of Figure 5.7.*

**Example 5.4.** *Consider the case where one of the model parameters drifts slowly online with*

$$G(q;t) = \frac{0.1159(q^3 + 0.5q)}{q^4 - 2.2q^3 + 2.42q^2 - \theta(t)q + 0.7225}, \quad \text{where } \theta(t) = \frac{1.87}{1 + t/1500}. \tag{5.9}$$

*The offline data length and the noise levels are $N = 50$ and $\sigma^2 = \sigma_p^2 = 0.01$, respectively. The SMM-PC algorithms are compared with the fixed signal matrix $\text{col}(U, Y)$ and the adaptive signal matrix $\text{col}(U_t, Y_t)$ with $\gamma = 1, 0.9, 0.7, 0.5$. The stage costs $J_t$ are plotted on the left of Figure 5.8, and the boxplot of the true total control cost $J_{\text{tot}}$ is shown on the right of Figure 5.8 with 50 Monte Carlo simulations.*

As can be seen from Examples 5.3, the control performance improves by using adaptive signal matrices as more online data accumulate. However, it cannot converge to the ideal MPC performance since the noise in $\mathbf{y}_{\text{ini}}$ still exists. For Example 5.4, the controller further benefits from a forgetting factor $\gamma < 1$ to reduce the weight of previous trajectories with less accurate model parameters. The best performance is achieved with $\gamma = 0.9$. These results demonstrate that incorporating online data effectively achieves better control performance for high noise levels and slowly time-varying systems.

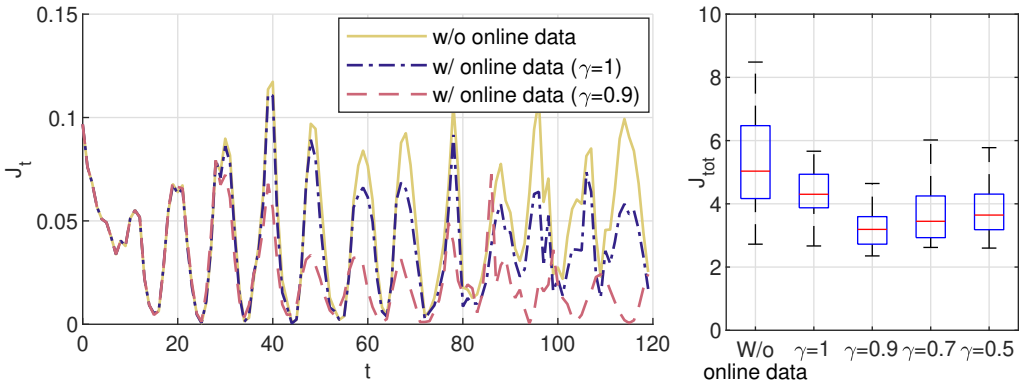



Figure 5.8: Effects of online data adaptation in SMM-PC for slowly time-varying systems. Left: stage cost $J_t$, right: boxplot of the true total control cost $J_{\text{tot}}$.

## 5.2 Stochastic Indirect Data-Driven Predictive Control

This section extends the certainty equivalence algorithm in the previous section by considering the predicted trajectory as a random vector whose statistics are specified in Section 4.4. In this section, we consider the state-space model (4.23) with disturbances, as discussed in Remarks 4.3 and 4.13. The following assumptions are adopted: 1) the noise in each column of $Y$ is independent (cf. Remark 4.12), 2) the uncertainties can be non-Gaussian (cf. Remark 4.11), and 3) the data-driven estimate of $\Gamma$ is correct, i.e., $\Gamma = \hat{\Gamma}_Z$. For completeness, results in Chapter 4 are restated under these conditions as follows, with slight abuse of notation.

Define $\Psi := \text{col}\,(U, W)$ and $\mathbf{w}^t := (w_k)_{k=t-L_0}^{t+L'-1}$. The distribution of the stochastic output prediction is given by

$$\mathbb{E}\left[\mathbf{y}^t \,|\, g^t\right] = \bar{\mathbf{y}}^t, \quad \text{cov}\left(\mathbf{y}^t \,|\, g^t\right) = \Sigma^t, \tag{5.10}$$

where

$$g^t := R_u \,\text{col}\left(\mathbf{u}_{\text{ini}}^t, \mathbf{u}^t\right) + R_w \mathbf{w}^t + R_2 \mathbf{y}_{\text{ini}}^t, \tag{5.11}$$

$$[R_u \; R_w] := F^{-1}\Psi^\top (\Psi F^{-1}\Psi^\top)^{-1}, \tag{5.12}$$

$$R_2 := \left(F^{-1} - F^{-1}\Psi^\top (\Psi F^{-1}\Psi^\top)^{-1}\Psi F^{-1}\right) Y_p^\top S, \tag{5.13}$$

$$\bar{\mathbf{y}}^t := Y_f g^t - \Gamma(Y_p g^t - \mathbf{y}_{\text{ini}}^t), \tag{5.14}$$

$$\Sigma^t := \left\|g^t\right\|_2^2 T + \Gamma \Sigma_{\text{yini}}^t \Gamma^\top + \Gamma_w \Sigma_w \Gamma_w^\top, \tag{5.15}$$

$$T := \sigma^2 \left(\Gamma\Gamma^\top + \mathbb{I}\right), \; \Gamma_w := (Y_f - \Gamma Y_p) R_w, \; \Gamma := Y_f R_2 \left(Y_p R_2\right)^{-1}. \tag{5.16}$$

Based on the stochastic predictor (5.10), the stochastic indirect DDPC algorithm can be proposed. In the following subsections, the stochastic control cost $J_{\text{ctr}}$ is first formulated as a quadratic objective. Then, the prediction accuracy is improved by filtering the output initial condition measurements with a Kalman filter. Finally, the output constraints are guaranteed to be satisfied

with high probability by formulating tightened SOC constraints.

### 5.2.1 Stochastic Control Cost

The stochastic control cost $J_{\text{ctr}}$ is formulated as a quadratic function in the following lemma.

**Lemma 5.1.** *The expected control cost is given by*

$$J_{\text{ctr}} = \left\|\mathbf{u}^t\right\|_{\bar{R}}^2 + \left\|\bar{\mathbf{y}}^t - \mathbf{r}^t\right\|_{\bar{Q}}^2 + \operatorname{tr}\left(\bar{Q}T\right)\left\|g^t\right\|_2^2 + \text{const.}, \tag{5.17}$$

*where* $\bar{R} := \mathbb{I}_{L'} \otimes R$, $\bar{Q} := \mathbb{I}_{L'} \otimes Q$, $\mathbf{r}^t := (r_{t+k})_{k=0}^{L'-1}$, *and* $T := \sigma^2\left(\Gamma\Gamma^\top + \mathbb{I}_{n_y L'}\right)$. *The cost is quadratic with respect to the optimization variable* $\mathbf{u}^t$.

*Proof.* The expected output cost is calculated as:

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{y}^t - \mathbf{r}^t\right\|_{\bar{Q}}^2\right] &= \mathbb{E}\left[\left(\bar{\mathbf{y}}^t + \mathbf{e}^t - \mathbf{r}^t\right)^\top \bar{Q}\left(\bar{\mathbf{y}}^t + \mathbf{e}^t - \mathbf{r}^t\right)\right] \\
&= \left\|\bar{\mathbf{y}}^t - \mathbf{r}^t\right\|_{\bar{Q}}^2 + \mathbb{E}\left[\left(\mathbf{e}^t\right)^\top \bar{Q}\mathbf{e}^t\right] = \left\|\bar{\mathbf{y}}^t - \mathbf{r}^t\right\|_{\bar{Q}}^2 + \operatorname{tr}\left(\bar{Q}\Sigma^t\right) \\
&= \left\|\bar{\mathbf{y}}^t - \mathbf{r}^t\right\|_{\bar{Q}}^2 + \operatorname{tr}\left(\bar{Q}T\right)\left\|g^t\right\|_2^2 + \text{const.},
\end{aligned}$$

where $\mathbf{e}^t$: $\mathbb{E}[\mathbf{e}^t] = \mathbf{0}$, $\operatorname{cov}(\mathbf{e}^t) = \Sigma^t$ is the prediction error. The second to last equality is due to the cyclic property of the trace function. This cost is quadratic with respect to $\mathbf{u}^t$ since both $g^t$ and $\bar{\mathbf{y}}^t$ are linear with respect to $\mathbf{u}^t$. □

The stochastic control cost adds a $\|g^t\|_2^2$-regularization term to the nominal cost. Such regularization is required in regularized DeePC (5.3) for well-definedness. However, in regularized DeePC, it is unclear how to tune the regularization parameter other than trial and error. By considering the regularizer as the uncertainty term in the expected output cost, the weighting factor can be reliably selected as $\operatorname{tr}\left(\bar{Q}T\right)$, which depends on the output cost matrix and the noise level.

### 5.2.2 Initial Condition Estimation

In model-based output-feedback MPC, an estimator has to be designed to estimate the initial state of the predictor, which is not measurable. This is not required in DDPC since the output initial condition $\mathbf{y}_{\text{ini}}^t$ can be directly measured. In fact, in most existing DDPC implementations with stochastic data, the output initial condition $\mathbf{y}_{\text{ini}}^t$ comes from measurements as in the deterministic case, i.e., $\mathbf{y}_{\text{ini}}^t := (y_k)_{k=t-L_0}^{t-1}$. Thus, the associated covariance $\Sigma_{\text{yini}}^t = \sigma_p^2\mathbb{I}$ is constant in (5.15). This source of uncertainty can be alleviated by choosing a larger $L_0$, as illustrated in the following example.

**Example 5.5.** *Consider the same trajectory data and online measurement noise as in Example 5.3. The controller performance of SMM-PC is compared with* $L_0 = n_x = 4$ *and* $L_0 = 10$. *As*

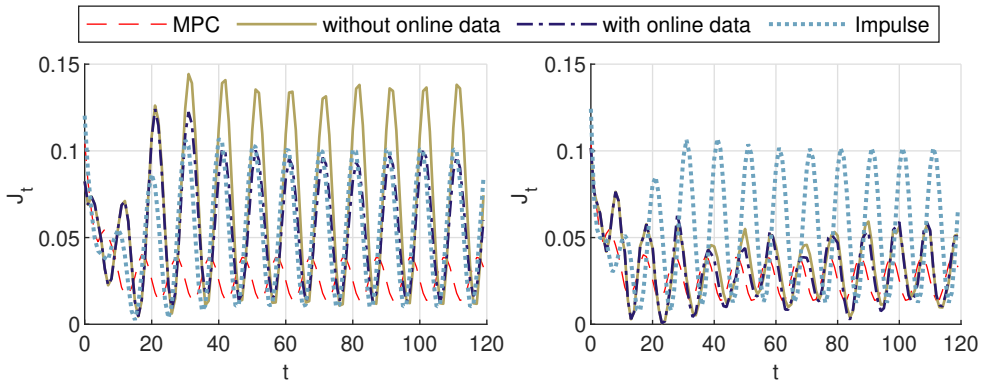



Figure 5.9: Stage costs with different $L_0$. Left: $L_0 = 4$, right: $L_0 = 10$.

*another comparison, an additional MPC algorithm is introduced with an FIR model obtained by simulating the SMM with a pulse, as discussed in Section 4.3.5. This impulse MPC algorithm does not require noisy past output measurements, so it circumvents the initial condition estimation problem. The stage costs $J_t$ are plotted in Figure 5.9.*

Results from Figure 5.9 demonstrate that when $L_0 = 4$, SMM-PC performs worse than impulse MPC without online data, and similarly when online data are incorporated, due to inaccurate initial condition measurements. When a larger value of $L_0 = 10$ is selected, SMM-PC performs significantly better than impulse MPC and is close to the ideal MPC algorithm.

On the other hand, in the presence of stochastic uncertainties, an adequately designed estimator in MPC can estimate the initial state with a diminishing covariance much smaller than the noise level in the measurements. Therefore, although not required, it can be beneficial to improve the output initial condition measurements by using output predictions at previous time steps to design an estimator, especially in cases where the online measurement error is significant. In this subsection, a Kalman filter is designed as the estimator. In particular, we replace $y_k$ with its Kalman-filtered counterpart as the output initial condition. This reduces the prediction errors by shrinking $\Sigma^t_{\text{yini}}$ as time progresses.

In detail, the same predictor (5.10) for predictive control design at time $(t-1)$ is used to filter the output at time $t$ and update $\mathbf{y}^t_{\text{ini}}$. The predictor can be considered as a non-minimal state-space "model" with "state"

$$\bar{x}_t := \text{col}\left(u_{t-L_0}, \ldots, u_{t-1}, y_{t-L_0,0}, \ldots, y_{t-1,0}\right). \tag{5.18}$$

Let $\bar{y}^t_k$ and $e^t_k$ denote the $(k+1)$-th block element of $\bar{\mathbf{y}}^t$ and $\mathbf{e}^t$, respectively, and $\Sigma^t_k$ be the covariance of $e^t_k$, i.e., the $(k+1)$-th $n_y \times n_y$ block on the diagonal of $\Sigma^t$. The data-driven "model"

is then given by

$$\begin{cases} \bar{x}_{t+1} &= \underbrace{\begin{bmatrix} \Lambda^{n_u} & \mathbf{0} \\ \mathbf{0} & \Lambda^{n_y} \end{bmatrix}}_{\bar{\Lambda}} \bar{x}_t + \begin{bmatrix} \mathbf{0} \\ u_0^t \\ \mathbf{0} \\ \bar{y}_0^t + e_0^t \end{bmatrix}, \\ \zeta_{t+1} &= \begin{bmatrix} \mathbf{0} & \mathbb{I}_{n_y} \end{bmatrix} \bar{x}_{t+1} + v_t = y_{t,0} + v_t = y_t, \end{cases} \tag{5.19}$$

where $\Lambda^k$ denotes the $k$-step upper shift matrix with ones on the $k$-th superdiagonal. The covariances of the "process noise" $e_0^t$ and the measurement noise $v_t$ are $\Sigma_0^t$ and $\sigma_p^2 \mathbb{I}_{n_y}$, respectively. Then, a Kalman filter for (5.19) can be designed to estimate the initial condition $\bar{x}_t$. Let the state estimate and the output part of the state error covariance be $\bar{x}_{t,t}$ and $P_{t,t}$, respectively. Then, the initial conditions for the DDPC problem can be set as $\mathrm{col}\left(\mathbf{u}_{\mathrm{ini}}^t, \mathbf{y}_{\mathrm{ini}}^t\right) := \bar{x}_{t,t}$ and $\Sigma_{\mathrm{yini}}^t := P_{t,t}$. The Kalman filtering algorithm is summarized in Algorithm 5.1.

**Algorithm 5.1** Kalman filter in stochastic indirect DDPC

1: **Initialization:**

$$\bar{x}_{0,0} := \mathrm{col}\left(u_{-L_0}, \ldots, u_{-1}, y_{-L_0}, \ldots, y_{-1}\right), \tag{5.20}$$
$$P_{0,0} := \mathbb{I}_{n_y L_0}. \tag{5.21}$$

2: **Prediction:**

$$\bar{x}_{t,t+1} := \bar{\Lambda} \bar{x}_{t,t} + \mathrm{col}\left(\mathbf{0}, u_0^t, \mathbf{0}, \bar{y}_0^t\right), \tag{5.22}$$
$$P_{t,t+1} := \Lambda^{n_y} P_{t,t} \left(\Lambda^{n_y}\right)^\top + \Sigma_0^t. \tag{5.23}$$

3: **Update:**

$$K_{t+1} := \Sigma_0^t \left(\Sigma_0^t + \sigma_p^2 \mathbb{I}_{n_y}\right)^{-1}, \tag{5.24}$$
$$\bar{x}_{t+1,t+1} := \bar{x}_{t,t+1} + \mathrm{col}\left(\mathbf{0}, K_{t+1}\left(y_t - \bar{y}_0^t\right)\right), \tag{5.25}$$
$$P_{t+1,t+1} := \left(\mathbb{I}_{n_y} - K_{t+1}\right) P_{t,t+1}. \tag{5.26}$$

**Remark 5.3.** *Only one-step-ahead prediction is required to run the Kalman filter. Here, it is obtained by truncating the same $L'$-step-ahead predictor used in predictive control for simplicity. One can also similarly construct a one-step-ahead data-driven predictor with $L' = 1$, specifically for Kalman filtering.*

**Remark 5.4.** *A similar idea was proposed in Alpago et al. (2020) for a direct DDPC algorithm. However, no approach is provided to quantify the covariance of the prediction error required in the Kalman filter as there is no well-defined predictor in direct DDPC.*

### 5.2.3 Chance Constraint Satisfaction

Due to the existence of unbounded noise, the output constraints $y_{t,0} \in \mathscr{Y}_t$ cannot be guaranteed robustly under unbounded noise. Instead, high-probability chance constraints are considered, either element-wise as

$$\Pr\left(h_i^{t+k\top} y_k^t \leq q_i^{t+k}\right) \geq p, \quad \forall i = 1,\dots,n_c,\ k = 0,\dots,L'-1, \tag{5.27}$$

or set-wise as

$$\Pr\left(y_k^t \in \mathscr{Y}_{t+k}\right) \geq p, \quad \forall k = 0,\dots,L'-1, \tag{5.28}$$

where $p$ is the targeted probability. These chance constraints are typically guaranteed by tightening the nominal constraints to account for prediction uncertainties. However, unlike standard model-based predictors with additive uncertainties, the prediction error covariance of the data-driven predictor (5.10) depends on the particular inputs and initial conditions via $g^t$. So, the amount of constraint tightening cannot be calculated offline. Define the augmented linear constraints by $\bar{\mathscr{Y}}_t := \{\mathbf{y} \,|\, \bar{H}^t \mathbf{y} \leq \bar{q}^t\}$, where

$$\bar{H}^t := \left[\bar{h}_1^t \ \dots \ \bar{h}_{L'n_c}^t\right]^\top := \text{blkdiag}\left(H^t,\dots,H^{t+L'-1}\right),$$
$$\bar{q}^t := \text{col}\left(\bar{q}_1^t,\dots,\bar{q}_{L'n_c}^t\right) := \text{col}\left(q^t,\dots,q^{t+L'-1}\right).$$

The following lemma guarantees chance constraint satisfaction by constraint tightening.

**Lemma 5.2.** *The constraint*

$$\bar{q}^t - \bar{H}^t \bar{\mathbf{y}}^t \geq \mu \sqrt{\text{diag}\left(\bar{H}^t \Sigma^t \bar{H}^{t\top}\right)} \tag{5.29}$$

*guarantees the satisfaction of the chance constraints* (5.27) *if* $\mu \geq \sqrt{\frac{1}{1-p} - 1}$ *and* (5.28) *if* $\mu \geq \sqrt{\frac{n_y}{1-p}}$.

*Proof.* Applying the one-sided Chebyshev's inequality, we have

$$\Pr\left(\bar{h}_i^t \mathbf{y}^t - \bar{h}_i^t \bar{\mathbf{y}}^t \leq \sqrt{\tfrac{1}{1-p} - 1} \cdot \text{std}\left(\bar{h}_i^t \mathbf{y}^t\right)\right) \geq p, \quad \forall i, \tag{5.30}$$

where $\text{std}\left(\bar{h}_i^t \mathbf{y}^t\right) = \sqrt{\bar{h}_i^t \Sigma^t \bar{h}_i^{t\top}}$. This leads to (5.27) for $\mu \geq \sqrt{\frac{1}{1-p} - 1}$ since $\bar{q}_i^t - \bar{h}_i^t \bar{\mathbf{y}}^t \geq \mu \sqrt{\bar{h}_i^t \Sigma^t \bar{h}_i^{t\top}}$ from (5.29).

From the multi-dimensional Chebyshev's inequality, the ellipsoidal set

$$\mathscr{E}_k^t := \left\{ e_k^t \,\middle|\, e_k^{t\top} \left(\Sigma_k^t\right)^{-1} e_k^t \leq \frac{n_y}{1-p} \right\} \tag{5.31}$$

is a confidence region of the prediction error $e_k^t$ with at least probability $p$. Then, the chance

constraint is satisfied if

$$\vec{y}_k^t \in \mathscr{Y}_{t+k} \ominus \mathscr{E}_k^t := \left\{ y \,\middle|\, y + e \in \mathscr{Y}_{t+k}, \forall e \in \mathscr{E}_k^t \right\}, \tag{5.32}$$

where $\ominus$ denotes the Pontryagin difference. For polytope $\mathscr{Y}_{t+k}$ and ellipsoid $\mathscr{E}_k^t$, we have

$$\mathscr{Y}_{t+k} \ominus \mathscr{E}_k^t = \left\{ y \,\middle|\, {h_i^{t+k}}^\top y \leq q_i^{t+k} - \eta_{\mathscr{E}_k^t}\left(h_i^{t+k}\right), i = 1, \ldots, n_c \right\}, \tag{5.33}$$

where $\eta_{\mathscr{E}_k^t}(h) := \sqrt{\frac{n_y}{1-p} h^\top \Sigma^t h}$ is the support function of $\mathscr{E}_k^t$. Aggregating the constraints for all $i$ leads to (5.28) for $\mu \geq \sqrt{\frac{n_y}{1-p}}$. □

**Remark 5.5.** *Let $F_{\chi^2(d)}(\cdot)$ and $F_{\mathcal{N}}(\cdot)$ be the cumulative distribution function of the $\chi^2$-distribution with $d$ degrees of freedom and the unit Gaussian distribution, respectively. The lemma can be tightened if Gaussian uncertainties are considered, i.e., both $v_t$ and $\mathbf{w}^t$ are Gaussian, by choosing $F_{\mathcal{N}}(\mu) \geq p$ for* (5.27) *and $F_{\chi^2(n_y)}(\mu^2) \geq p$ for* (5.28)*. The proof is very similar to that of Lemma 5.2.*

Unfortunately, the tightened constraint (5.29) is not convex. The following corollary provides a convex surrogate for (5.29).

**Corollary 5.1.** *The second-order cone (SOC) constraint*

$$\vec{q}^t - \bar{H}^t \bar{\mathbf{y}}^t \geq \mu \left( \mathbf{c}_1 + \mathbf{c}_2 \left\| g^t \right\|_2 \right), \tag{5.34}$$

*where*

$$\mathbf{c}_1 := \sqrt{\operatorname{diag}\left( \bar{H}^t \left( \Gamma \Sigma_{\text{yini}}^t \Gamma^\top + \Gamma_w \Sigma_w \Gamma_w^\top \right) {\bar{H}^t}^\top \right)}, \tag{5.35}$$

$$\mathbf{c}_2 := \sqrt{\operatorname{diag}\left( \bar{H}^t T {\bar{H}^t}^\top \right)}, \tag{5.36}$$

*guarantees the satisfaction of* (5.29)*.*

*Proof.* Since $\sqrt{\sum_i a_i} \leq \sum_i \sqrt{a_i}$, we have $\mathbf{c}_1 + \mathbf{c}_2 \left\| g^t \right\|_2 \geq \sqrt{\mathbf{c}_1^2 + \mathbf{c}_2^2 \left\| g^t \right\|_2^2} = \sqrt{\operatorname{diag}\left( \bar{H}^t \Sigma^t {\bar{H}^t}^\top \right)}$. □

The proposed stochastic indirect DDPC algorithm is summarized in Algorithm 5.2.

**Remark 5.6.** *Algorithm 5.2 is only rigorously applicable to linear systems. However, it has been demonstrated numerically (Dörfler et al., 2023) and experimentally (Elokda et al., 2021) that such DDPC algorithms can be applied to nonlinear systems as well. Nonlinearities can be considered as additional output errors in the signal matrix $Z$. To estimate the magnitude of the combined output error of both the output noise and the nonlinearities, linear system identification can be conducted to set the noise level $\sigma^2$ to the MSE of the prediction with the linear model.*

**Algorithm 5.2** Stochastic indirect DDPC

1: Select a data-driven predictor and calculate predictor parameters from (5.12), (5.13), and (5.16).
2: Initialize the Kalman filter from Algorithm 5.1.
3: **for** $t = 0$ **to** $N_c - 1$ **do**
4: **begin**
5: $\mathrm{col}\,(\mathbf{u}^t_{\mathrm{ini}}, \mathbf{y}^t_{\mathrm{ini}}) \leftarrow \bar{x}_{t,t}$, $\Sigma^t_{\mathrm{yini}} \leftarrow P_{t,t}$, update $\mathbf{w}^t$.
6: $\mathbf{u}^t \leftarrow \underset{\mathbf{u}^t}{\mathrm{argmin}}$ (5.17) s.t. (5.11),(5.14),(5.34), $u^t_k \in \mathscr{U}_{t+k}$, $\forall\, k = 0, \ldots, L' - 1$.
7: Apply the first entry in the optimized input trajectory $u_t = u^t_0$ to the system and measure $y_t$.
8: Run the Kalman filter from Algorithm 5.1.
9: **end**

### 5.2.4 Numerical Results

This subsection compares the performance of nominal indirect DDPC (5.4) (*N-DDPC*), indirect DDPC with initial condition estimation in Algorithm 5.1 (*KF-DDPC*), and stochastic indirect DDPC in Algorithm 5.2 (*S-DDPC*). Consider the following fourth-order dynamics:

$$\left[\begin{array}{c|c|c} A & B & E \\ \hline C & 0 & 0 \end{array}\right] = \left[\begin{array}{cccc|c|c} 0.36 & 0.64 & 0.07 & 0.02 & 0.29 & 0.03 \\ 0.42 & 0.58 & 0.02 & 0.07 & 0.03 & 0.20 \\ -9.34 & 9.34 & 0.23 & 0.58 & 4.90 & 1.07 \\ 5.88 & -5.88 & 0.39 & -0.39 & 1.07 & 3.48 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{5.37}$$

The following parameters are used in the example: $L_0 = 4$, $L' = 10$, $Q = 20$, $R = 1$, $\sigma^2 = \sigma_p^2 = 0.01$, $\mathbf{w}^t = \mathbf{0}$, $\Sigma_w = 0.001 \cdot \mathbb{I}$, and $p = 0.95$. An offline trajectory of length 500 is collected with unit Gaussian inputs, and the signal matrix is constructed with a Hankel structure, which leads to $M = 487$. The disturbances are sampled from an i.i.d. Gaussian distribution of variance 0.001. The elementwise chance constraints (5.27) are used. The same online noise and disturbance sequences are used to compare the three algorithms. The minimum-MSE predictor is employed as the predictor. No input constraint is considered in this example, i.e., $\mathscr{U}_t = \mathbb{R}$. Upper and lower output bounds are specified as the output constraints.

The closed-loop trajectories of the algorithms are presented in Figure 5.10, alongside the reference trajectory and the output bounds. As observed in Figure 5.10, *KF-DDPC* outperforms *N-DDPC* by introducing the initial condition estimator, although constraint violations are still evident. *S-DDPC* further enhances *KF-DDPC*, particularly in terms of constraint satisfaction. To underscore the effectiveness of the Kalman filter, Figure 5.11 showcases the comparison between the filtered output initial conditions and the measured ones for *S-DDPC*. The filtered trajectory is notably closer to the true trajectory than the measured trajectory.

The performance is further evaluated quantitatively by 50 Monte Carlo simulations with different

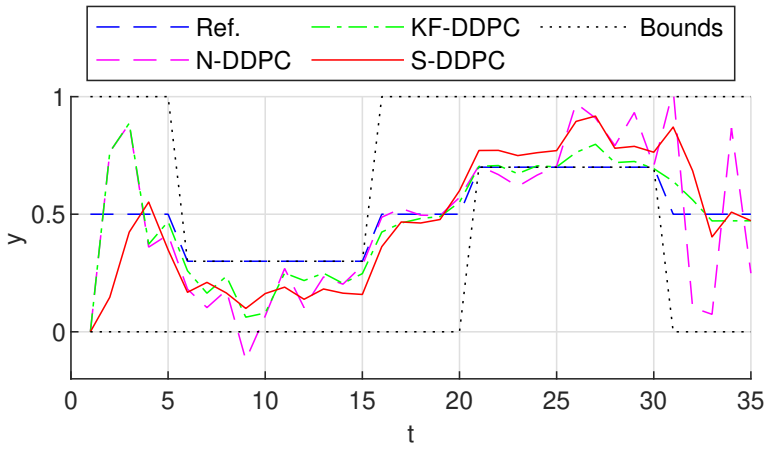



Figure 5.10: Closed-loop trajectories of indirect DDPC algorithms.

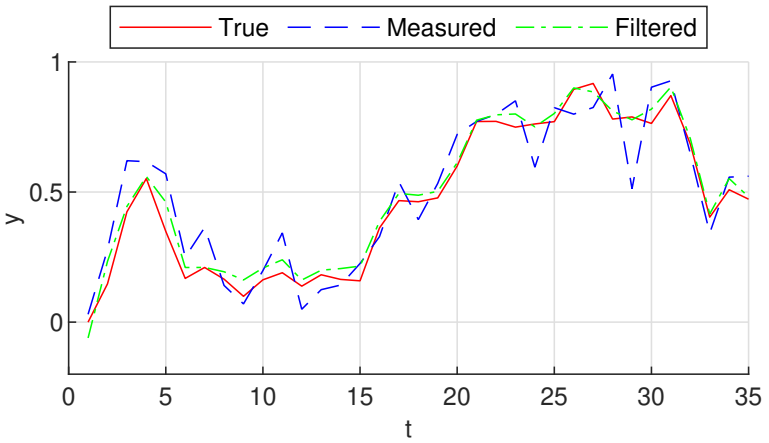



Figure 5.11: Comparison of the filtered and measured output trajectories.

noise and disturbance realizations. Figure 5.12 shows the boxplots of the true total control cost $J_{\text{tot}}$ and the total amount of constraint violations, calculated as $\sum_t \max\left(H^t y_{t,0} - q^t, 0\right)$. The results validate our observations from Figure 5.10 that our proposed algorithm *S-DDPC* performs much better than the nominal algorithm with almost no constraint violation.

## 5.3 High-Fidelity Simulation Results: Space Heating Control

Finally, we present high-fidelity simulation results tested on a space heating control case study to evaluate the performance of the proposed stochastic indirect DDPC algorithm against other DDPC algorithms in practice.

The system considered in this study is a three-room apartment known as the urban mining and recycling (UMAR) unit in the NEST research building of the Swiss Federal Laboratories for Material Science and Technology (Empa) in Dübendorf, Switzerland. The unit's layout is

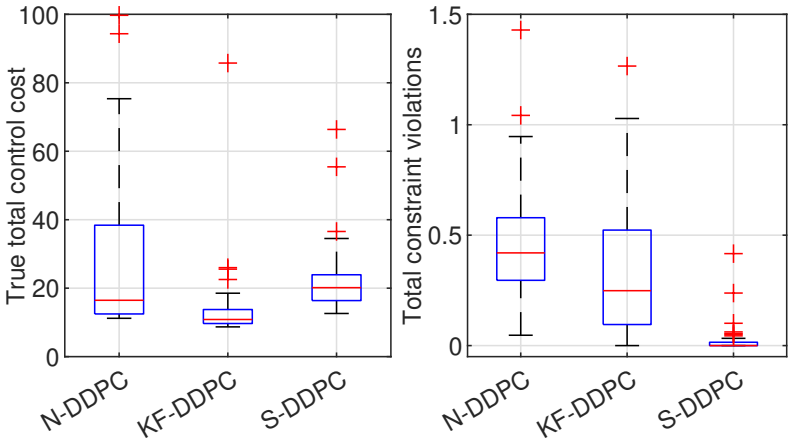



Figure 5.12: Boxplots of (a) the true total control cost $J_{\text{tot}}$ and (b) the total amount of constraint violations.


Figure 5.13: Layout of the UMAR unit with the controlled rooms marked. © Werner Sobek.

illustrated in Figure 5.13. Each room is treated as one thermal zone, and the heating power is dissipated into the zones through constant volume radiant ceiling panels. The continuous power set point, determined by the controller, is realized by regulating the valve opening, which is converted into discrete opening and closing sequences using pulse-width modulation. Additionally, the space heating control is subject to constraints imposed by occupants' perception of comfort. Thermal comfort bounds are expressed as predefined temperature limits in this study. As the unit is residential, the unit is considered to be unoccupied during the day and occupied during the night. The definition leads to relaxed temperature constraints during unoccupied hours. Specifically, the constraints are set to be between 20 °C and 26 °C from 08:00 to 16:59 and between 22 °C and 24 °C from 17:00 to 07:59 of the following day.

The performance of multiple control algorithms is benchmarked on a high-fidelity white-box EnergyPlus model *nestli* of the UMAR unit, developed in Khayatian et al. (2022); Bojarski et al. (2023). The model is constructed based on detailed knowledge of the system layout and the

heat storage and transfer characteristics of the construction materials. In addition, the model is calibrated with three-year field measurements. This provides a digital twin of the system for accurately benchmarking different control algorithms. To actively override variables during the simulation, the EnergyPlus model is wrapped into a functional mock-up unit and integrated into a Simulink simulation model.

MATLAB & Simulink simulations are conducted to implement space heating control of the three main rooms (Rooms 272, 273, and 274 in Figure 5.13) in the UMAR unit using the MATLAB version of the *nestli* model. The control sample time is 15 minutes. The three rooms are separately controlled. For each room, the controller is implemented as follows.

**Input**. The heating power [kW] is considered as the input $u_t$ and is constrained to be positive and upper-bounded by the maximum heating power. The maximum heating power depends on the difference between the supply and the return temperatures [°C]. Since the heating system is not modeled in this study, constant temperature differences are considered throughout the control horizon. The input command is implemented by pulse-width modulation of binary valve positions with a 1-minute resolution.

**Output**. The room temperature [°C] is considered as the output $y_t$. Gaussian output noise with a variance of 0.01 is added to all measurements.

**Disturbance**. The ambient temperature [°C] and the global horizontal irradiance [W/m$^2$] are considered as the disturbances $w_t$. The variances of the ambient temperature and the irradiance predictions are assumed to be 0.04 and 25, respectively.

**Objective**. The control objective is to minimize the total energy consumption within the control horizon, i.e., $J_{\text{ctr}} = \|\mathbf{u}^t\|_1$.

The following control algorithms are compared in simulation.

***SMM-PC***. Algorithm 5.2 is applied using the minimum-MSE predictor but without the Kalman filter.

***N4SID***. A state-space model of order $n_x = L_0$ is identified by subspace identification in the innovation form:

$$\begin{cases} \hat{x}_{t+1} &= \hat{A}\hat{x}_t + \hat{B}u_t + \hat{E}w_t + \hat{L}e_t, \\ y_t &= \hat{C}\hat{x}_t + \hat{D}u_t + e_t, \end{cases} \tag{5.38}$$

using the MATLAB command `n4sid`, where $\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$, $\hat{E}$, and $\hat{L}$ are the identified model parameters, $\hat{x}_t$ is the state estimate, and $e_t$ quantifies the model error whose variance is also estimated by `n4sid`. Stochastic MPC with chance constraints described in Oldewurtel et al. (2012) is implemented by considering both the stochastic model error $e_t$ and the disturbance uncertainty in $w_t$. This is a representative control algorithm using the classical system identification paradigm.

***BiLevel***. The robust bi-level DDPC algorithm presented in Lian et al. (2023) is implemented. The

algorithm augments the standard indirect DDPC (5.4) with robustness to bounded disturbance uncertainties and an affine disturbance feedback term (Oldewurtel et al., 2008). Since stochastic disturbance uncertainties are considered in this work, a high-probability disturbance set w.p. $p$ is used as the disturbance bound. Note that the online adaptation in Lian et al. (2023) is not implemented for a fair comparison against other algorithms.

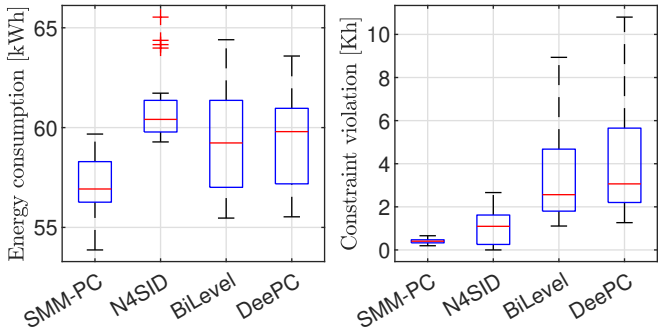
***DeePC***. The direct DDPC algorithm described in (16) of Dörfler et al. (2023) is implemented with $p = 2$. Nominal output constraints are enforced since this approach cannot quantify the prediction error due to its ill-defined predictor structure.

The following parameters are used for the controllers: $L_0 = 10$ (2.5 h), $L' = 15$ (3.75 h), and $p = 0.7$. Optimal control problems are solved by Gurobi. In addition, a hysteresis controller is considered as the baseline.

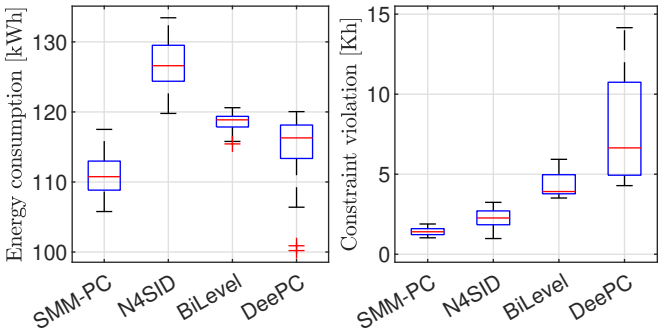
Historical weather data from November 7 to December 7, 2020, are used to run the simulations. The offline data used to construct the signal matrix $Z$ are collected from November 7 to November 30, 2020, by running the default hysteresis controller. This corresponds to 2'305 data points. The data-driven controllers are applied from December 1 to December 7, 2020, after an initialization phase of 2.5 hours corresponding to the length of $L_0$.

Monte Carlo simulations of 20 runs are conducted for each algorithm with different realizations of measurement noise and disturbance uncertainties. All the considered predictive control algorithms perform significantly better than the baseline hysteresis control, with a 20%–27% reduction in energy consumption and a 30%–93% reduction in constraint violation. So, in what follows, we focus on benchmarking different predictive control algorithms. Results of energy consumption and temperature constraint violation for the three controlled rooms are shown in Figure 5.14. It can be seen that the *SMM-PC* algorithm outperforms the other algorithms in terms of both energy consumption and constraint satisfaction for all three rooms. Specifically, compared to *N4SID*, *BiLevel*, and *DeePC*, *SMM-PC* reduces the constraint violation by 59%, 77%, and 90% with an average energy saving of 8%, 6%, and 4%, respectively. A comparison of the closed-loop input-output trajectories in one representative simulation is shown in Figure 5.15. One can observe that *SMM-PC* is more conservative in regulating the room temperature to avoid constraint violation. It is worth mentioning that the control decisions of *SMM-PC* are also smoother than other algorithms, which implies reduced hardware wear in practice.

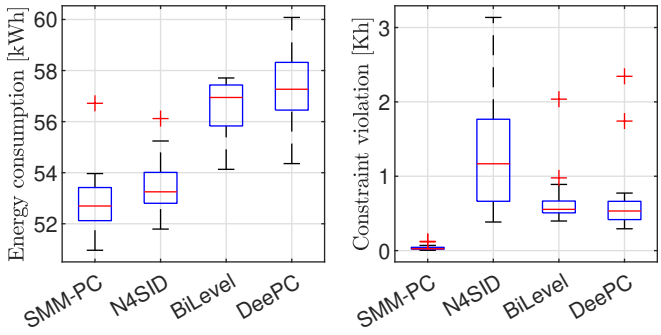
The *SMM-PC* algorithm is analyzed more closely in Figure 5.16, where the one-step-ahead predictions and the tightened bounds of *SMM-PC* are also plotted for Room 272 as an example. The tightened bounds for *N4SID*, calculated offline based on an estimate of the model error, are also compared. The predicted room temperature is very close to the true room temperature, and the tightened bounds are more conservative than *N4SID*, leading to fewer constraint violations. The same plot for *BiLevel* is shown in Figure 5.17. Although the prediction accuracy is also acceptable, *BiLevel* underestimates the prediction error, leading to more constraint violations. This is expected since *BiLevel* is only robust to disturbance prediction errors but not measurement




(a) Room 272




(b) Room 273

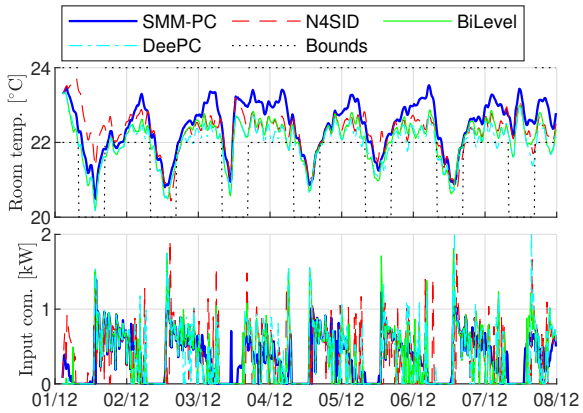



(c) Room 274

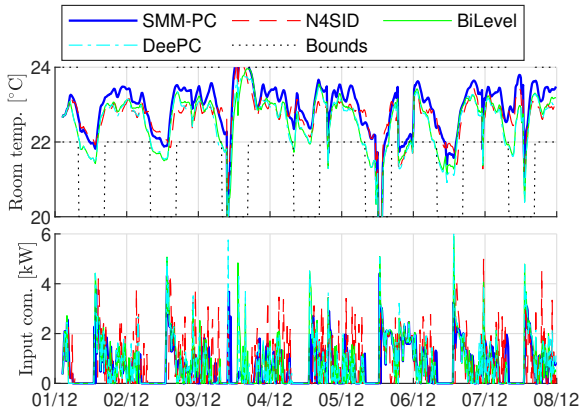
Figure 5.14: Boxplots of energy consumption and constraint violation for different predictive control algorithms.

errors.

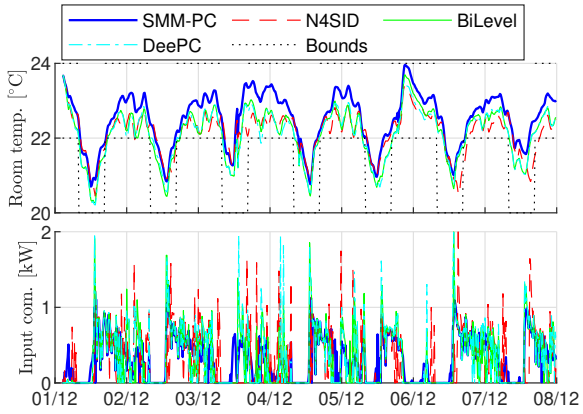
In the results presented above, we selected $\lambda_g = 10^4$ for *DeePC*, which is the best-performing one in our tests with $\lambda_g = 10^i$, $i = 2,3,4,5,6$. For this $\lambda_g$ choice, the behaviors of *BiLevel* and *DeePC* are similar as shown in Figure 5.15, except that there is no constraint tightening for *DeePC*. This validates the statement in Dörfler et al. (2023) that direct and indirect DDPC methods are equivalent for sufficiently large $\lambda_g$ with nominal predictions. However, the *DeePC* algorithm may completely fail for small $\lambda_g$ values. Figure 5.18 illustrates one scenario where *DeePC* cannot provide reasonable temperature control for Room 273 with $\lambda_g = 10^2$.




(a) Room 272




(b) Room 273




(c) Room 274

Figure 5.15: Representative input-output trajectories of different predictive control algorithms.

Finally, to further evaluate the reliability of the algorithms, two scenarios with deteriorated uncertainties are considered: 1) the temperature measurements are less accurate with a ten times larger output noise variance, and 2) the disturbance prediction is less accurate with ten times larger

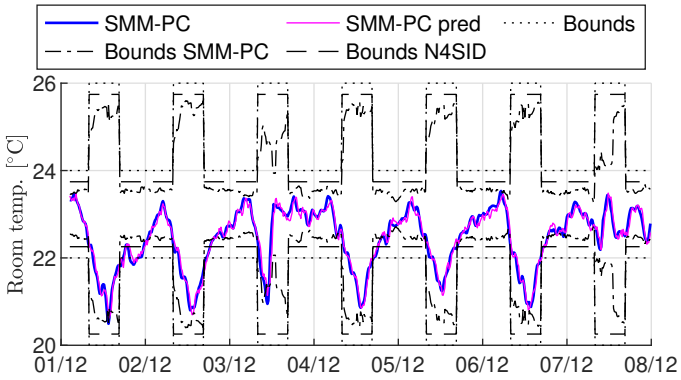



Figure 5.16: Prediction accuracy and constraint tightening of *SMM-PC* in Room 272.

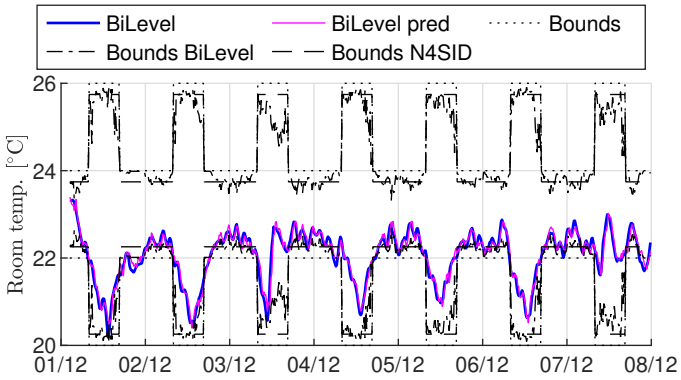



Figure 5.17: Prediction accuracy and constraint tightening of *BiLevel* in Room 272.

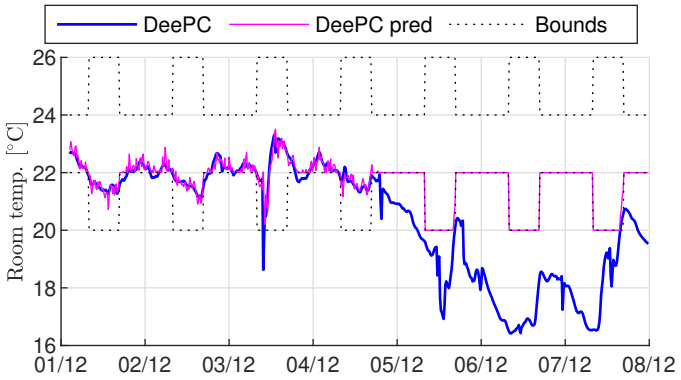



Figure 5.18: Malfunctioning of *DeePC* with $\lambda_g = 100$ in Room 273.

Table 5.2: Energy consumption and constraint violation results of different algorithms under high uncertainties. Scenario 1: high output noise, scenario 2: high disturbance prediction errors. Values in brackets indicate changes with respect to the nominal results.

(a) Scenario 1: average energy consumption [kWh]

| | **Room 272** | **Room 273** | **Room 274** |
|---|---|---|---|
| **SMM-PC** | 58.1 (+1.1) | 112.4 (+1.4) | 52.4 (-0.5) |
| **N4SID** | 63.3 (+2.2) | 132.0 (+5.3) | 57.0 (+3.6) |
| **Bilevel** | 61.5 (+2.0) | 127.5 (+8.9) | 55.8 (-0.7) |
| **DeePC** | 59.4 (+0.0) | 120.2 (+5.8) | 54.4 (-2.8) |

(b) Scenario 2: average energy consumption [kWh]

| | **Room 272** | **Room 273** | **Room 274** |
|---|---|---|---|
| **SMM-PC** | 59.6 (+2.6) | 107.1 (-4.0) | 53.2 (+0.4) |
| **N4SID** | 61.1 (+0.0) | 129.7 (+3.0) | 53.5 (+0.1) |
| **Bilevel** | 60.0 (+0.5) | 126.1 (+7.5) | 57.0 (+0.5) |
| **DeePC** | 59.5 (+0.1) | 117.7 (+3.2) | 57.2 (-0.1) |

(c) Scenario 1: average constraint violation [°Ch]

| | **Room 272** | **Room 273** | **Room 274** |
|---|---|---|---|
| **SMM-PC** | 0.25 (-0.16) | 1.69 (+0.29) | 0.06 (+0.03) |
| **N4SID** | 0.25 (-0.73) | 1.62 (-0.58) | 0.47 (-0.83) |
| **Bilevel** | 3.46 (+0.25) | 10.45 (+6.10) | 3.82 (+3.15) |
| **DeePC** | 3.06 (-1.15) | 5.62 (-2.36) | 1.79 (+1.12) |

(d) Scenario 2: average constraint violation [°Ch]

| | **Room 272** | **Room 273** | **Room 274** |
|---|---|---|---|
| **SMM-PC** | 0.04 (-0.37) | 1.49 (+0.08) | 0.04 (+0.00) |
| **N4SID** | 1.05 (+0.07) | 2.02 (-0.17) | 1.28 (-0.02) |
| **Bilevel** | 2.49 (-0.71) | 3.33 (-1.02) | 0.46 (-0.21) |
| **DeePC** | 4.00 (-0.21) | 7.54 (-0.44) | 0.67 (+0.00) |

variances for both the ambient temperature and the irradiance. The Monte Carlo simulations are repeated for these two scenarios, and the results are shown in Table 5.2. It can be observed that *SMM-PC* remains the best regarding both energy consumption and constraint violation in both scenarios. In addition, the performance of *SMM-PC* in these two scenarios is very consistent compared to the nominal case despite the increased uncertainties. In contrast, an increase in energy consumption in Room 273 is observed for the other algorithms. The performance of the *BiLevel* algorithm deteriorates significantly in scenario 1, as the algorithm cannot handle large output noise.

## 5.4 Summary

This chapter applies nonparametric data-driven predictors to receding horizon predictive control. By adopting a linearized signal matrix model predictor with certainty equivalence, the signal matrix model predictive control algorithm demonstrates superior performance compared to subspace predictive control and regularized data-enabled predictive control with the possibility to incorporate online data for improving poor offline data or learning parameter drifts.

The algorithm is then extended to a stochastic control framework with several modifications based on prediction error characterization. These modifications provide a tuning-free regularizer design in the control cost, improved initial condition estimation, and reliable constraint satisfaction, which are achieved by evaluating the expected cost, designing a Kalman filter, and formulating convex constraint tightening terms, respectively.

The results are verified in high-fidelity simulations of a space heating control example. The proposed stochastic indirect data-driven predictive control algorithm achieves constraint satisfaction more reliably with less energy consumption than existing data-driven predictive control methods.

# Identification of Periodic Systems Part III

# 6 Identification of Linear Time-Periodic Systems

This chapter discusses the problem of identifying linear time-periodic (LTP) systems in both the time and frequency domains. Lifting and switching methods are available to convert LTP models to equivalent LTI models for applying LTI identification methods, which is discussed in Section 6.1. However, the reverse conversion is not trivial. Additional requirements should be satisfied such that the identified LTI model is realizable to its LTP form.

In the time domain, the atomic norm regularization approach discussed in Chapter 2 is extended to LTP system identification in Section 6.2 for low-complexity estimation, which uses the switching reformulation of LTP systems. In this case, the main structural requirement is that the LTI sub-models should have a consistent model order throughout the period with the exact pole locations, referred to as the *uniformity* requirement in what follows. This requirement is difficult to enforce with existing methods but can be satisfied in the atomic norm regularization framework with a group lasso regularizer introduced in 2.1.1. In particular, model parameters corresponding to the same atomic dynamics with the same poles are grouped across LTI sub-models. A case study and Monte Carlo simulations show that the proposed method effectively estimates uniform low-complexity LTP models and is superior to existing methods in model fitting under low SNR's.

In the frequency domain, based on early work from Wereley (1990), harmonic transfer function (HTF) coefficients are identified in Louarroudi et al. (2012); Yin and Mehr (2009); Allen and Sracic (2009); Shin et al. (2005) using least-square methods. This approach leads to high-order nonparametric models. Non-convex optimization methods are used to directly identify state-space models in Goos and Pintelon (2016), with the drawback that globally optimal estimates are not guaranteed to be obtained. On the other hand, subspace methods are widely applied to identifying LTP systems (Wood et al., 2018; Liu, 1997; Felici et al., 2007), but the majority of works use the time-domain approach described in Verhaegen and Yu (1995) for LTV systems. A frequency-domain subspace method is proposed by Uyanik et al. (2019) based on frequency lifting. However, the method is limited to SISO systems with multi-sinusoidal inputs. In addition, the frequency grid must be specially designed to avoid overlaps between different periodic harmonics, and model order selection can be problematic under low SNR's.

Section 6.3 presents an alternative frequency-domain subspace method that is compatible with more general inputs and MIMO systems by extending McKelvey et al. (1996) using time-domain lifting. First, the frequency response of the lifted system is identified by generalized empirical transfer function estimation (ETFE). By utilizing the frequency response estimates of the lifted system, the time-aliased periodic impulse response of the original LTP system can be obtained by a linear map. This leads to an order-revealing decomposition of LTP systems with a block Hankel structure, followed by a conventional subspace routine that identifies the range space of the extended observability matrix by SVD. This algorithm is proven to be consistent under a general class of output noise. Compared to Uyanik et al. (2019), the main advantages are that it can be applied to MIMO systems and that generic periodic inputs can be used. However, compared with previous time-domain methods that use arbitrary input-output data sequences, this method requires an ensemble of periodic identification data that are harmonic with the fundamental frequency of the system. Finally, the proposed algorithm is compared to time-domain methods by numerical simulation to show its advantage when using periodic identification data. The consistency property is also verified in simulation.

## 6.1 LTP Systems and Their LTI Reformulations

The theory of LTP systems, as well as their LTI reformulations, are briefly reviewed in this section. See Bittanti and Colaneri (2009) for more detailed explanations.

Consider a strictly causal and stable discrete-time LTP system:

$$\begin{cases} x_{t+1} &= A_t x_t + B_t u_t, \\ y_t &= C_t x_t + v_t, \end{cases} \tag{6.1}$$

where $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$, $y_t \in \mathbb{R}^{n_y}$, $v_t \in \mathbb{R}^{n_y}$ are the states, inputs, and outputs, and output noise, respectively. The time-varying matrices $A_t = A_{t+T^\star}$, $B_t = B_{t+T^\star}$, $C_t = C_{t+T^\star}$ are periodic state-space matrices of appropriate dimensions, where $T^\star$ is the period length, which is assumed known. The stability of LTP systems can be assessed by the spectral radius of the monodromy matrix $\Psi_{A,\tau} := A(\tau-1)A(\tau-2)\dots A(\tau-T^\star)$. Bittanti (1986) proved that the eigenvalues of $\Psi_{A,\tau}$ are independent of $\tau$ and that the system is stable iff the spectral radius $\rho(\Psi_{A,\tau}) < 1$. Denote the collection of unique $A$-matrices as $\bar{A} := \operatorname{col}\left(A_0^\top, A_1^\top, \dots, A_{T^\star-1}^\top\right)$, and similarly for $\bar{B}$ and $\bar{C}$. The periodic impulse response of the system is defined as

$$g_r^t := C_t A_{t-1} A_{t-2} \dots A_{t-r+1} B_{t-r} \in \mathbb{R}^{n_y \times n_u}, \tag{6.2}$$

where the superscript $t$ denotes the current tag time and the subscript $r > 0$ denotes the time difference between the input and the output. Since the dynamics are periodic, $g_r^t$ is also $T^\star$-periodic with respect to $t$. The extended controllability and observability matrices of LTP systems

are defined as

$$\mathcal{C}_s^\tau := \begin{bmatrix} B_{\tau-1} & A_{\tau-1}B_{\tau-2} & \cdots & A_{\tau-1}\ldots A_{\tau-s+1}B_{\tau-s} \end{bmatrix} \in \mathbb{R}^{n_x \times sn_u}, \tag{6.3}$$

$$\mathcal{O}_s^\tau := \begin{bmatrix} C_\tau \\ C_{\tau+1}A_\tau \\ \vdots \\ C_{\tau+s-1}A_{\tau+s-2}\ldots A_\tau \end{bmatrix} \in \mathbb{R}^{sn_y \times n_x}, \tag{6.4}$$

respectively.

### 6.1.1 Lifting and Switching

Lifting and switching are two main reformulations of LTP systems for applying LTI methods. The lifting method converts the LTP system to a structured $T^\star$-times slower LTI system with $T^\star$-times larger input and output dimensions (Bittanti and Colaneri, 2000). The state dimension remains the same. In the lifted system, the inputs and outputs of one whole period in the LTP system are concatenated as the augmented inputs and outputs:

$$\tilde{u}_k := \operatorname{col}\left(u_{kT^\star}, u_{kT^\star+1}, \ldots, u_{kT^\star+T^\star-1}\right), \tag{6.5}$$

and similarly for $\tilde{y}_k$ and $\tilde{v}_k$. The augmented dynamics thus become:

$$\begin{cases} \tilde{x}_{k+1} &= \Psi_{A,0}\tilde{x}_k + \bar{\mathcal{C}}_{T^\star}^0 \tilde{u}_k, \\ \tilde{y}_k &= \mathcal{O}_{T^\star}^0 \tilde{x}_k + \Xi\, \tilde{u}_k + \tilde{v}_k, \end{cases} \tag{6.6}$$

where $\tilde{x}_k := x_{kT^\star}$ is the lifted state,

$$\bar{\mathcal{C}}_s^\tau := \begin{bmatrix} A_{\tau-1}\ldots A_{\tau-s+1}B_{\tau-s} & A_{\tau-1}\ldots A_{\tau-s+2}B_{\tau-s+1} & \cdots & B_{\tau-1} \end{bmatrix} \in \mathbb{R}^{n_x \times sn_u} \tag{6.7}$$

denotes the flipped controllability matrix, and

$$\Xi := \begin{bmatrix} \mathbb{0} & \mathbb{0} & \mathbb{0} & \cdots & \mathbb{0} \\ g_1^1 & \mathbb{0} & \mathbb{0} & \cdots & \mathbb{0} \\ g_2^2 & g_1^2 & \mathbb{0} & \cdots & \mathbb{0} \\ \vdots & \vdots & \vdots & \ddots & \mathbb{0} \\ g_{T^\star-1}^{T^\star-1} & g_{T^\star-2}^{T^\star-1} & g_{T^\star-3}^{T^\star-1} & \cdots & \mathbb{0} \end{bmatrix} \tag{6.8}$$

indicates the feedthrough term within the period. Due to its natural connection to the subspace identification formulation, the method is usually used to extend the subspace identification method (Verhaegen and Yu, 1995).

In the switching method, the LTP system is reformulated as a switched LTI system with $T^\star$

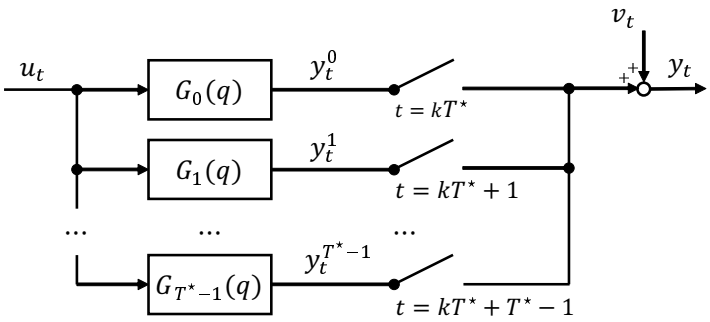



Figure 6.1: Illustration of the switching reformulation of LTP systems.

switches. In detail, the system (6.1) is expressed with the following input-output model:

$$y_t = \sum_{l=1}^{\infty} g_l^t u_{t-l} + v_t. \tag{6.9}$$

For a fixed $t$, $\{g_l^t\}_{l=1}^{\infty}$ formulates a valid IIR model of an LTI system as

$$G_\tau(q) := \sum_{l=1}^{\infty} g_l^\tau q^{-l} := C_\tau (q^{T^\star}\mathbb{I} - \Psi_{A,\tau})^{-1} \mathcal{B}_\tau(q), \tag{6.10}$$

where

$$\mathcal{B}_\tau(q) := \sum_{i=0}^{T^\star-1} A_{\tau-1}A_{\tau-2}\ldots A_{\tau+i-T^\star+1}B_{\tau+i}\cdot q^i, \tag{6.11}$$

$q$ is the forward time-shift operator, and $\tau = 0, 1, \ldots, T^\star - 1$. This is known as the periodic transfer function of the system (6.1). The models $G_\tau(q)$ are called *sub-models* in the following. Thus, a periodically switched LTI model of the LTP system can be defined as

$$y_t = y_t^\tau, \quad t = kT^\star + \tau, \quad \text{where } y_t^\tau = G_\tau(q)u_t + v_t. \tag{6.12}$$

See Figure 6.1 for a diagrammatic illustration. Note that the dynamics of each switch $G_\tau(q)$ have precisely the same poles, which are the solutions to $f_{\Psi,\tau}\left(q^{T^\star}\right) = 0$, where $f_{\Psi,\tau}(x)$ is the characteristic polynomial of $\Psi_{A,\tau}$. The solutions are independent of $\tau$ because they are the $T^\star$-th roots of the eigenvalues of the monodromy matrix, which are independent of $\tau$. Therefore, for a uniform LTP system (6.1), the poles in each sub-model are the same. The model order of the switched sub-systems is then $T^\star n_x$. This reformulation has been used to estimate HTF's in Yin and Mehr (2009).

Comparing both methods, lifting constructs a system that is $T^\star \times T^\star$-times larger than the original system, whereas switching decomposes the system into $T^\star$ sub-systems of the same size. The additional parameters induced by redundant dimensions in lifting are constrained by structural constraints on the lifted systems, such as causality. These constraints are generally difficult to enforce in identification but can be exploited in subspace identification as demonstrated in

Section 6.3. Switching, conversely, preserves the input-output dimensions of the LTP system at the expense of augmented model orders. This problem is alleviated by using regularization techniques as shown in Section 6.2, where the computational complexity does not scale with the model order, as a sufficiently high-order model structure is adopted at the beginning anyway.

## 6.2 Low-Order Regularization of LTP Systems

In this section, we consider the problem of identifying a uniform low-order switched model (6.10) of a SISO LTP system from a sequence of input-output data $(u_t, y_t)_{t=1}^{N_p T^\star}$, where $N_p$ is the number of periods observed.

Similar to Chapter 3, the input-output model (6.9) is truncated to a FIR model at a sufficiently high order, denoted by $n_g$, for tractability. Then, a least squares problem can be formulated to identify the sub-models independently by minimizing the squared $l_2$-norm of the prediction error:

$$\min_{\bar{\mathbf{g}}} \quad J_{\mathrm{LS}}\left(\bar{\mathbf{g}} | (u_t, y_t)_{t=1}^{N_p T^\star}\right) := \sum_{\tau=1}^{T^\star} \sum_{k=0}^{N_p-1} \left[ y_{kT^\star+\tau} - \sum_{l=1}^{n_g} g_l^\tau u_{kT^\star+\tau-l} \right]^2, \tag{6.13}$$

where

$$\bar{\mathbf{g}} := \begin{bmatrix} \bar{\mathbf{g}}^1 & \cdots & \bar{\mathbf{g}}^{T^\star} \end{bmatrix} := \begin{bmatrix} g_1^1 & g_1^2 & \cdots & g_1^{T^\star} \\ g_2^1 & g_2^2 & \cdots & g_2^{T^\star} \\ \vdots & \vdots & \ddots & \cdots \\ g_{n_g}^1 & g_{n_g}^2 & \cdots & g_{n_g}^{T^\star} \end{bmatrix} \in \mathbb{R}^{n_g \times T^\star} \tag{6.14}$$

gathers the parameters in all sub-models.

However, this unregularized problem does not enforce the requirement that the identified system should be uniform and low-order. This section investigates the extension of low-order regularizers to LTP systems. We first discuss the common rank regularizer to see why it is not suitable for LTP system identification. Then, it is shown that a grouped version of atomic norm regularization can effectively regularize the estimator to be uniform and low-order.

### 6.2.1 Rank Regularization

One of the most common low-order regularizers is based on the fact that the rank of the extended observability and controllability matrices gives the McMillan degree of the system. Different matrices that reveal this rank for regularization have been constructed. In the switched IIR model

(6.9), the Hankel operator on the impulse responses

$$\mathcal{H}_L(\bar{\mathbf{g}}^t) := \begin{bmatrix} g_1^t & g_2^t & \cdots & g_{n_g-L+1}^t \\ g_2^t & g_3^t & \cdots & g_{n_g-L+2}^t \\ \vdots & \vdots & \ddots & \vdots \\ g_L^t & g_{L+1}^t & \cdots & g_{n_g}^t \end{bmatrix} \tag{6.15}$$

is commonly used as the rank-revealing matrix. As also mentioned in Section 4.3.2, since the rank function is highly non-convex, its best convex surrogate, the nuclear norm, is applied in optimization for tractability as in Smith (2014); Fazel et al. (2001). Thus, we have the following convex nuclear norm regularizer:

$$\mathcal{R}_{\mathrm{N}}(\bar{\mathbf{g}}) := \sum_{\tau=1}^{T^\star} \beta_\tau \left\| \mathcal{H}_L(\bar{\mathbf{g}}^\tau) \right\|_*, \tag{6.16}$$

where $\beta := \mathrm{col}\,(\beta_1, \ldots, \beta_{T^\star})$ is the weighting vector to control sub-model complexity. Note that the nuclear norm can be seen as the generalization of the $l_1$-norm to matrices.

However, besides its general issue of stability (Pillonetto et al., 2016) and scalability (Shah et al., 2012), the Hankel nuclear norm regularizer fails to provide an explicit expression for the model order. This makes it hard to tune different sub-models to the same order, not to mention the requirement of the same pole locations. As demonstrated in Section 6.2.3, this regularizer often cannot regularize the sub-systems to any given order despite fine-tuning the weighting vector $\beta$.

### 6.2.2 Grouped Atomic Norm Regularization

In this subsection, the atomic norm regularization discussed in Chapter 2 is adopted on sub-models to enforce the uniformity requirement while overcoming the stability and scalability issues of the Hankel nuclear norm regularization. Similar to the LTI case, the sub-models are decomposed as linear combinations of atoms:

$$G_\tau(q) = \sum_{k \in \mathcal{K}} c_k^\tau A_k(q) \approx \sum_{i=1}^{p} c_{k_i}^\tau \cdot A_{k_i}(q) =: \mathbf{c}_\tau^T \mathbf{A}(q), \tag{6.17}$$

where the infinite atom set is approximated by fine gridding $(k_i)_{i=1}^p$ with a vector of atoms $\mathbf{A}(q) := \mathrm{col}\left(A_{k_1}(q), A_{k_2}(q), \ldots, A_{k_p}(q)\right)$. The vector $\mathbf{c}_\tau := \mathrm{col}\left(c_{k_1}^\tau, c_{k_2}^\tau, \ldots, c_{k_p}^\tau\right) \in \mathbb{C}^p$ denotes the corresponding coefficients, and $p$ is the number of atoms in the grid.

Then the orders of the sub-models are equal to the cardinality of $\mathbf{c}_\tau$. Using the $l_1$-norm as the surrogate for the cardinality function, the atomic norm of the system $G_\tau(q)$ with respect to $\mathbf{A}(q)$ is defined as $\|\mathbf{c}_\tau\|_1$. It was shown in Shah et al. (2012) that the atomic norm is a good approximation of the Hankel nuclear norm.

As discussed in Section 2.2.1, for real-valued systems, additional constraints on $\mathbf{c}_\tau$ are required:

$$c_{k_i}^\tau = \text{conj}\left(c_{k_j}^\tau\right), \quad \forall k_i = \text{conj}(k_j),\ \tau = 1,2,\ldots,T^\star. \tag{6.18}$$

To apply atomic norm regularization to the combined switched model, the expansion (6.17) is rewritten in terms of the impulse response matrix as

$$\bar{\mathbf{g}} = \bar{\mathbf{g}}^a \mathbf{c}, \tag{6.19}$$

where $\bar{\mathbf{g}}^a := [\bar{\mathbf{g}}_1^a\ \bar{\mathbf{g}}_2^a\ \ldots\ \bar{\mathbf{g}}_p^a] \in \mathbb{R}^{n_g \times p}$, $\bar{\mathbf{g}}_i^a$ is the $n_g$-truncated impulse response of $A_{k_i}(q)$, and $\mathbf{c} := [\mathbf{c}_1\ \mathbf{c}_2\ \ldots\ \mathbf{c}_{T^\star}] \in \mathbb{C}^{p \times T^\star}$. Note that $\bar{\mathbf{g}}^a$ is a constant matrix that can be pre-computed. Thus, we have the following atomic norm regularized optimization problem:

$$\min_{\mathbf{c}}\ J_{\text{LS}}\left(\bar{\mathbf{g}}^a\mathbf{c}\middle|(u_t,y_t)_{t=1}^{N_pT^\star}\right) + \lambda\mathcal{R}_{\text{A}}(\mathbf{c}), \quad \text{s.t. } (6.18), \tag{6.20}$$

where

$$\mathcal{R}_{\text{A}}(\mathbf{c}) := \sum_{\tau=1}^{T^\star} \beta_\tau \|\mathbf{c}_\tau\|_1 \tag{6.21}$$

is the atomic norm regularizer which is the weighted sum of the atomic norms of the sub-models.

The atomic norm regularizer $\mathcal{R}_{\text{A}}(\mathbf{c})$ guarantees the smoothness and stability of the estimated system. In addition, problem (6.20) is a QP problem, which has much better scalability compared to the semidefinite programming (SDP) problem induced by the nuclear norm regularization. Although the uniformity requirement is still not guaranteed since each sub-model is separately regularized, the pole location information is now accessible from the estimated parameters $\mathbf{c}$. This information can be used to propose a uniform regularizer that guarantees the same pole locations for each sub-model.

The basic idea to modify the previous LTI-based atomic norm regularizer (6.21) to satisfy the uniformity requirement is to connect the same atom at different tag times. The same atom needs to be either included in all or excluded from all of the sub-model dynamics. To do this, we first examine the structure of the parameter matrix $\mathbf{c}$. If the $(i,j)$-th element in $\mathbf{c}$ is non-zero, the sub-model $j$ has a pole at $k_i$ and vice versa. Therefore, in addition to the sparsity requirement induced by the low-order assumption, each row of $\mathbf{c}$ also needs to be either all zero or all non-zero. This requirement coincides with the concept of grouping in group lasso by considering each row as a group. So, the following grouped atomic norm regularizer is proposed:

$$\mathcal{R}_{\text{GA}}(\mathbf{c}) := \sum_{i=1}^{p} \left\|\mathbf{c}^{(i)}\right\|_2, \tag{6.22}$$

in place of $\mathcal{R}_{\text{A}}(\mathbf{c})$ in (6.20), where $\mathbf{c}^{(i)}$ denotes the $i$-th row of $\mathbf{c}$. There is only one hyperparameter $\lambda$ for this regularizer instead of a vector of hyperparameters $\beta_\tau$ for the previous two regularizers. In this work, hyperparameters are selected by cross-validation with additional validation data $(u_t^v, y_t^v)_{t=1}^{N_vT^\star}$.

The algorithm for LTP system identification with grouped atomic norm regularization is summarized in Algorithm 6.1.

**Algorithm 6.1** LTP system identification with grouped atomic norm regularization

**Input:** $T^\star$, $(u_t, y_t)_{t=1}^{N_p T^\star}$, $(u_t^v, y_t^v)_{t=1}^{N_v T^\star}$
Select $n_g$, $(k_i)_{i=1}^{p}$ and compute $\bar{\mathbf{g}}^a$.
**for** $\lambda = \lambda_1$ **to** $\lambda_{n_\lambda}$ **do**
    **begin**
    $\mathbf{c}(\lambda) \leftarrow \underset{\mathbf{c}}{\operatorname{argmin}}\, J_{\mathrm{LS}}\left(\bar{\mathbf{g}}^a \mathbf{c} \middle| (u_t, y_t)_{t=1}^{N_p T^\star}\right) + \lambda \mathcal{R}_{\mathrm{GA}}(\mathbf{c})$, s.t. (6.18)
    $\varepsilon(\lambda) \leftarrow J_{\mathrm{LS}}\left(\bar{\mathbf{g}}^a \mathbf{c}(\lambda) \middle| (u_t^v, y_t^v)_{t=1}^{N_v T^\star}\right)$
    **end**
$\lambda^\star \leftarrow \underset{\lambda}{\operatorname{argmin}}\, \varepsilon(\lambda)$
**Output:** $\mathbf{c}^\star = \mathbf{c}(\lambda^\star)$, $\bar{\mathbf{g}}^\star = \bar{\mathbf{g}}^a \mathbf{c}(\lambda^\star)$

**Remark 6.1.** *A similar grouping concept can be extended to MIMO systems, where sub-models are defined as SISO FIR models for each input-output channel at each tag time. Similarly, the same atom in all these sub-models should have consistent sparsity and thus be grouped.*

### 6.2.3 Numerical Results

This section compares the grouped atomic norm method with other LTP system identification schemes. First, a variable-length pendulum system is examined to show the effectiveness of the grouped atomic norm method in estimating uniform low-order models, in contrast to other low-order methods. Furthermore, Monte Carlo simulation demonstrates that the proposed method fits the true system better than other regularized methods. It also outperforms the time-domain subspace identification method under low SNR's.

The following five identification schemes for LTP systems are compared. The first four methods use the switched FIR model of order $n_g = 100$. The least squares method (*LS*) solves the problem (6.13). The Hankel nuclear norm method (*Hank*) solves the regularized least squares problem with the regularizer (6.16). The Hankel matrices are constructed with $L = 20$. The atomic norm method (*Atom*) solves the problem (6.20). Our proposed method, the grouped atomic norm method (*GAtom*), modifies the problem (6.20) with the grouped regularizer (6.22). The atom set used in *Atom* and *GAtom* is defined by the poles $k = \alpha \exp(j\beta)$, where $\alpha = [0.02 : 0.02 : 0.98\ 0.99\ 0.999]$, $\beta = 0 : \pi/50 : \pi$ in the MATLAB notation, as suggested in Pillonetto et al. (2016). This gives a total of $p = 2601$ poles. The widely-used subspace identification method (*Sub*) proposed in Verhaegen and Yu (1995) is also compared, where the model order determined by singular value truncation is selected by cross-validation over a uniform order grid between 2 to 10. This method uses the lifting reformulation.

The optimization problems are solved by MOSEK. In terms of computational time, *LS* and *Sub* are the fastest by solving unconstrained least squares problems; *Hank* is slower than *Atom* and

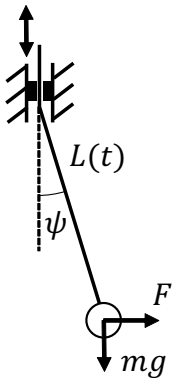



Figure 6.2: Illustration of the variable-length pendulum system.

*GAtom* because of its SDP nature. The difference is more significant as the period length $T^\star$ and the data length $N_pT^\star$ increase.

**Case study of a variable-length pendulum.** Consider a variable-length pendulum shown in Figure 6.2 with a periodic length profile $L(t) = L_0 + l\cos\omega t$. The non-linear dynamics of the system are given by

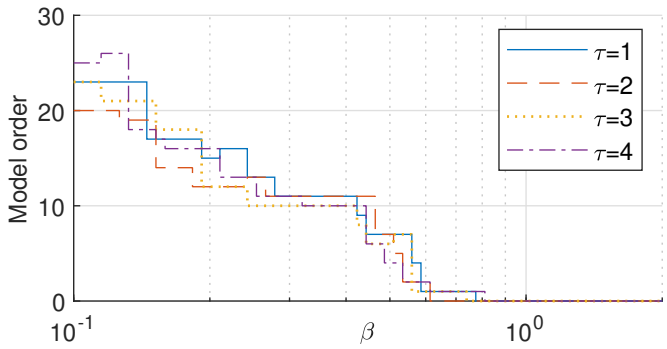
$$\ddot{\psi} = -\frac{g}{L(t)}\sin\psi + \frac{2\omega l\sin\omega t}{L(t)}\dot{\psi} + \frac{1}{mL(t)}F\cos\psi. \tag{6.23}$$

The following parameters are used: $L_0 = 10$, $l = 5$, $m = 5$, $g = 9.8$, and $\omega = 4\pi$. This system is modeled as a discrete-time SISO LTP system at small $\psi$, with $F$ as the input and $\psi$ as the output. The period length $T^\star$ is selected as 4 with a sampling time of $T_s = 2\pi/(T^\star\omega)$. A data set of length $N_pT^\star = 500$ is simulated with unit Gaussian inputs $u_t \sim \mathcal{N}(0,1)$ and Gaussian output noise $v_t \sim \mathcal{N}(0,\sigma^2)$, $\sigma^2 = (0.1\pi/180)^2$ for identification.

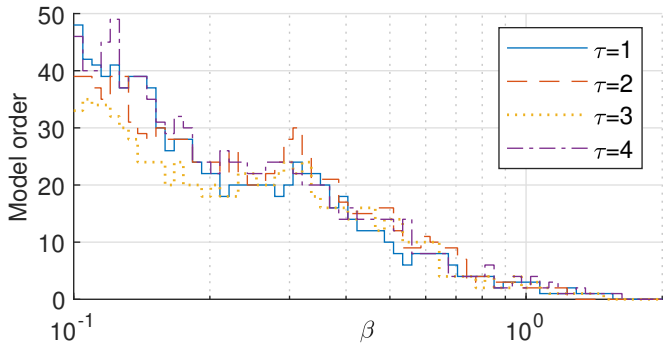
First, we try to obtain a uniform low-order model by fine-tuning sub-model complexity coefficients $\beta_\tau$ in *Hank* and *Atom*. In this example, $\beta_\tau$ is selected from a 100-point logarithmic grid between $10^{-1}$ and $10^1$, and $\lambda$ is fixed to 1. The relations between the $\beta_\tau$ values and the estimated model orders are shown in Figure 6.3. It can be seen that since the model order is indirectly controlled by coefficients $\beta_\tau$ with no explicit expression, the sub-models cannot be regularized to any given order for both *Hank* and *Atom*. This makes it hard to tune the sub-model orders to be uniform, especially as $T^\star$ increases. In contrast, *GAtom* always gives a uniform estimation for any choice of the scalar hyperparameter $\lambda$ with the same grid, as shown in Figure 6.4. These uniform models can then be selected by cross-validation.

**Monte Carlo simulation.** To compare the fitting performance of the proposed method with other methods, a Monte Carlo test campaign is set up as follows.

A bank of 100 low-order discrete-time SISO LTP systems of period length $T^\star = 2$ is generated. The model orders are randomly selected between 2 and 10. Continuous-time dynamics at each tag




(a) *Hank*




(b) *Atom*

Figure 6.3: Estimated sub-model orders with sub-model complexity tuning.

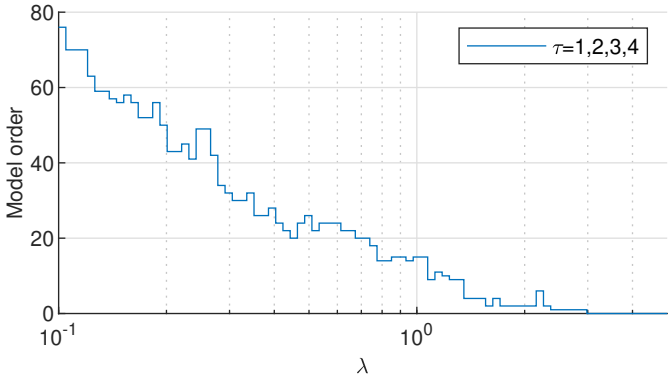



Figure 6.4: Estimated sub-model orders with *GAtom*.

time are generated by the MATLAB function `rss`. These continuous-time systems are sampled at three times their bandwidths and discretized by zero-order hold equivalence. They are also normalized to have a steady-state gain of 1. The resulting LTP systems are verified to be stable.

The systems are excited by unit Gaussian inputs $u_t \sim \mathcal{N}(0,1)$. The outputs are perturbed at two different output noise levels with $v_t \sim \mathcal{N}(0,\sigma^2)$, $\sigma^2 = 0.1, 0.01$. The initial states of the systems are set to 0. Two data sets of length $N_p T^\star = 500$ are generated for identification and validation,

respectively. The hyperparameter $\lambda$ in the regularized methods is cross-validated over a 10-point logarithmic grid between $10^{-1}$ and $10^{1}$. The sub-model complexity is not tuned ($\beta_\tau = \mathbf{1}$) for *Hank* and *Atom*, as this tuning is complicated to automate and often impractical to unify the orders of sub-models as can be seen from the case study.

Similar to (1.8), the performance of the estimators is assessed by comparing to the true model with the following fitting metric:

$$W := 100 \cdot \left( 1 - \left[ \frac{\sum_{\tau=1}^{T^\star} \sum_{i=1}^{n_g} (g_i^\tau - \hat{g}_i^\tau)^2}{\sum_{\tau=1}^{T^\star} \sum_{i=1}^{n_g} (g_i^\tau - \bar{g})^2} \right]^{1/2} \right), \tag{6.24}$$

where $g_i^\tau$ are the true impulse response coefficients in model (6.9), $\hat{g}_i^\tau$ are the estimated coefficients, and $\bar{g}$ is the mean of the true coefficients. The impulse response coefficients of the state-space model obtained by *Sub* are calculated by (6.2) for performance comparison.

The results of Monte Carlo simulation are demonstrated by statistics in Table 6.1 and boxplots in Figure 6.5, under the low ($\sigma^2 = 0.01$) and the high ($\sigma^2 = 0.1$) noise levels. It is shown that under both noise levels, the *LS* method cannot give satisfactory estimates. Under the high noise level, the *LS* estimation even fails to provide any information about the system with a negative average fitting. Comparing the three regularized methods, our proposed *GAtom* method achieves the best model fitting by incorporating the requirement on pole locations. *Atom* performs better than *Hank* due to its guaranteed stability.

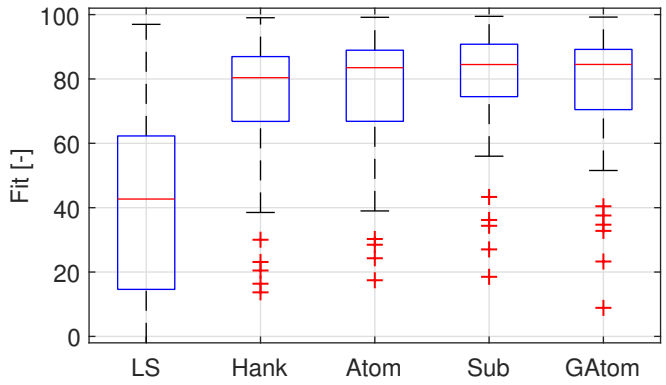
The *Sub* method has an advantage over *GAtom* under the low noise level with a higher mean fitting and a lower standard deviation. This is because the subspace identification gives a consistent estimator that converges to the true value in the noise-free case, whereas regularized methods are generally inconsistent. Nevertheless, the advantage of *GAtom* in model fitting is observed under the high noise level.

Table 6.1: Statistics of fitting performance.

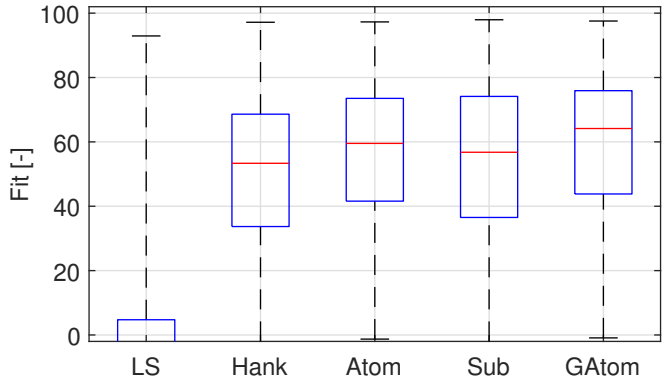
| | $\sigma^2 = 0.01$ | | | | | $\sigma^2 = 0.1$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **LS** | **Hank** | **Atom** | **Sub** | **GAtom** | **LS** | **Hank** | **Atom** | **Sub** | **GAtom** |
| **Mean** | 21.4 | 68.6 | 70.6 | 78.8 | 71.7 | -106.1 | 47.8 | 52.6 | 48.7 | 55.6 |
| **Median** | 42.7 | 80.4 | 83.5 | 84.5 | 84.5 | -79.5 | 53.3 | 59.5 | 56.8 | 64.2 |
| **Std** | 86.2 | 37.8 | 38.5 | 22.6 | 39.9 | 203.5 | 36.5 | 36.2 | 48.3 | 34.1 |

## 6.3 Frequency-Domain Subspace Identification of LTP Systems

In this section, we consider the problem of estimating a state-space LTP model that is equivalent to (6.1) up to a similarity transform. A total of $J$ input-output data sequences are collected with periodic inputs of length $N_p T^\star$, where $J \geq n_u T^\star$. The inputs, outputs, and output noise of the $i$-th




(a) $\sigma^2 = 0.01$




(b) $\sigma^2 = 0.1$

Figure 6.5: Comparison of fitting performance under different noise levels.

data sequence are denoted by $u_t^i$, $y_t^i$, and $v_t^i$ with the lifted versions indicated by $\tilde{u}_k^i$, $\tilde{y}_k^i$, and $\tilde{v}_k^i$, respectively. The following mild assumptions on the output noise are considered in this section.

**Assumption 6.1.** *The output noise is zero mean and stationary with fast-decaying covariances,*

$$\sum_{\tau=1}^{\infty} \left| \tau \cdot \mathbb{E}\left( \tilde{v}_{k,p}^i \tilde{v}_{k-\tau,p}^i \right) \right| \leq c_p, \tag{6.25}$$

*where $\tilde{v}_{k,p}^i$ is the $p$-th element of $\tilde{v}_k^i$ and $c_p$ is a finite constant. The noise is also i.i.d. across different data sequences and not correlated with the inputs.*

To develop a frequency-domain subspace identification method for LTP systems based on the frequency response of the lifted system, we first examine the subspace algorithm for LTI systems. This is briefly summarized as follows based on the uniformly-spaced data case in McKelvey et al. (1996).

Suppose $M$ frequency response data points $G_k \in \mathbb{C}^{n_y \times n_u}$, $k = 0, 1, \ldots, M-1$ of an LTI system are given on uniformly-spaced frequencies $\omega_k = 2\pi k/M$. First, apply the inverse discrete Fourier

transform (IDFT) on $G_k$,

$$h_t := \frac{1}{M}\sum_{k=0}^{M-1} G_k \cdot \exp\left(j\frac{2\pi tk}{M}\right),\ t = 1,2,\ldots,M. \tag{6.26}$$

The sequence $h_t$ is then the *time-aliased* impulse response of the system,

$$h_t = \sum_{i=0}^{\infty} g_{t+iM}. \tag{6.27}$$

Based on this result, the block Hankel matrix of $h_t$ has the following decomposition that reveals the order of the system:

$$H := \begin{bmatrix} h_1 & h_2 & \cdots & h_r \\ h_2 & h_3 & \cdots & h_{r+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_q & h_{q+1} & \cdots & h_{r+q-1} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{q-1} \end{bmatrix} (\mathbb{I} - A^M)^{-1} \begin{bmatrix} B & AB & \cdots & A^{r-1}B \end{bmatrix}, \tag{6.28}$$

where $\text{rank}(H) = n_x$ if $qn_y, rn_u \geq n_x$. Thus, the extended observability matrix of the system can be identified up to a similarity transform from the range space of $H$ by SVD and truncation. The order of the estimated system can be determined by thresholding or cross-validation.

This method is extended to the lifting reformulation of LTP systems in the following subsections.

### 6.3.1 Frequency Response of Lifted LTP Systems

An important characteristic of LTP systems is that, unlike LTI systems, an input with spectral content at frequency $\omega$ would generate an output response not only at $\omega$, but also at a series of other harmonics $\omega + 2k\pi/T^\star,\ k \in \mathbb{Z}$ (Wereley, 1990). Thus, the frequency response at a particular frequency $\omega$ is not a complex gain but a function $G_\omega(\omega + 2k\pi/T^\star)$ of $k$. This function-valued frequency response can be estimated at individual frequencies with a technique known as frequency lifting (Uyanik et al., 2019). However, this method is very restrictive in input design in that only carefully designed multi-sinusoidal inputs can be applied to ensure no overlap of harmonics with different input frequency contents. This work considers a time-lifted method for arbitrary periodic inputs of length $N_pT^\star$. As will be seen in Section 6.3.2, this method helps extend the available frequency-domain subspace identification algorithm to LTP systems.

In particular, the frequency response matrix of the lifted LTI system $G(\mathrm{e}^{j\omega_k})$ is used as the frequency response data of the original LTP system. It is shown in Section 4.3 of Bittanti and Colaneri (2000) that the frequency response of the lifted system is given by

$$G_{l,m}(\mathrm{e}^{j\omega_k}) = \sum_{s=0}^{\infty} g^l_{sT^\star+l-m} \exp(-j\omega_k s), \tag{6.29}$$

where $G_{l,m}(\mathrm{e}^{j\omega_k}) \in \mathbb{C}^{n_y \times n_u}$ denotes the $(l,m)$-th block element of $G(\mathrm{e}^{j\omega_k})$ and $\omega_k = \frac{2\pi k}{N_p}$, $k = 0,1,\ldots,N_p-1$. The frequency dependence may be omitted for simplicity in what follows. Due to the strict causality assumption of (6.1), $g_r^t = 0$ for all non-strictly-causal impulse response coefficients, that is, for $r \leq 0$.

Despite its LTI structure, frequency response estimation of the lifted MIMO system is not a trivial problem as conventional methods such as swept-sine and multi-sines (Dobrowiecki et al., 2006) do not apply to lifted LTP systems as the input channels cannot be excited separately, since they come from the same input sequence. Therefore, we first present the following generalized ETFE $\hat{G}(\mathrm{e}^{j\omega_k})$ from an ensemble of time-domain identification data with periodic inputs.

Apply the discrete Fourier transform (DFT) on the lifted inputs and outputs of each experiment:

$$U_i(\mathrm{e}^{j\omega_k}) := \sum_{n=0}^{N_p-1} \tilde{u}_n^i \exp\left(-j\frac{2\pi nk}{N_p}\right), \tag{6.30}$$

and similarly for $Y_i(\mathrm{e}^{j\omega_k})$. Then, the frequency response estimate is given as

$$\hat{G}(\mathrm{e}^{j\omega_k}) := \tilde{Y}(\mathrm{e}^{j\omega_k})\tilde{U}^{\dagger}(\mathrm{e}^{j\omega_k}), \tag{6.31}$$

where

$$\tilde{U}(\mathrm{e}^{j\omega_k}) := \begin{bmatrix} U_1(\mathrm{e}^{j\omega_k}) & U_2(\mathrm{e}^{j\omega_k}) & \ldots & U_J(\mathrm{e}^{j\omega_k}) \end{bmatrix}, \tag{6.32}$$

and similarly for $\tilde{Y}(\mathrm{e}^{j\omega_k})$. Here, for the right pseudoinverse to be well defined, $\tilde{U}(\mathrm{e}^{j\omega_k})$ needs to have full row rank, which requires $J \geq n_u T^\star$.

The estimate (6.31) generalizes the ETFE for the SISO case

$$\hat{G}(\mathrm{e}^{j\omega_k}) = \frac{Y(\mathrm{e}^{j\omega_k})}{U(\mathrm{e}^{j\omega_k})} \tag{6.33}$$

with multiple experiments to satisfy the persistency of excitation requirement for MIMO systems. Lemma 6.1 shows that this estimate has similar properties to the ETFE, i.e., it is unbiased with bounded covariances, and the estimation errors are independent across different frequencies. Note that for notational simplicity, a multiple-input single-output (MISO) structure is considered in the proof, but the same properties hold for the MIMO system with the covariance of the vectorized $\hat{G}(\mathrm{e}^{j\omega_k})$.

**Lemma 6.1.** *Under Assumption 6.1, the frequency response estimate (6.31) has the following properties:*

1. $\mathbb{E}\left[\hat{G}(\mathrm{e}^{j\omega_k})\right] = G(\mathrm{e}^{j\omega_k})$*;*
2. $\mathrm{cov}\left[\hat{G}^{(p)}\right] = \left(\Phi_{v_p} + \rho_p(N_p)\right)\left(\tilde{U}^{\dagger}\right)^{\mathsf{H}}\tilde{U}^{\dagger}$*, where* $\hat{G}^{(p)}$ *denotes the p-th row of* $\hat{G}$*,* $\Phi_{v_p}$ *is the power spectral density of the p-th element of* $\tilde{v}_k^i$*, and* $|\rho_p(N_p)| \leq 2c_p/N_p$*; and*
3. *estimates at different frequencies are independent.*

*Proof.* Define the DFT of the output noise sequences as $\tilde{V}(\mathrm{e}^{j\omega_k})$ similar to $\tilde{U}(\mathrm{e}^{j\omega_k})$ and $\tilde{Y}(\mathrm{e}^{j\omega_k})$. Decompose the lifted MIMO system into $n_y T^\star$ MISO systems with

$$G(\mathrm{e}^{j\omega_k}) =: \mathrm{col}\left(G^{(1)}(\mathrm{e}^{j\omega_k}), G^{(2)}(\mathrm{e}^{j\omega_k}), \ldots, G^{(n_y T^\star)}(\mathrm{e}^{j\omega_k})\right), \tag{6.34}$$

and similarly for $\hat{G}(\mathrm{e}^{j\omega_k})$. Then, with periodic inputs, we have

$$\tilde{Y}^{(p)}(\mathrm{e}^{j\omega_k}) = G^{(p)}(\mathrm{e}^{j\omega_k})\tilde{U}(\mathrm{e}^{j\omega_k}) + \tilde{V}^{(p)}(\mathrm{e}^{j\omega_k}), \tag{6.35}$$

$$\hat{G}^{(p)}(\mathrm{e}^{j\omega_k}) = G^{(p)}(\mathrm{e}^{j\omega_k}) + \tilde{V}^{(p)}(\mathrm{e}^{j\omega_k})\tilde{U}^\dagger(\mathrm{e}^{j\omega_k}), \tag{6.36}$$

where $\tilde{Y}^{(p)}(\mathrm{e}^{j\omega_k})$, $\tilde{V}^{(p)}(\mathrm{e}^{j\omega_k})$ denote the $p$-th row of $\tilde{Y}(\mathrm{e}^{j\omega_k})$, $\tilde{V}(\mathrm{e}^{j\omega_k})$, respectively. With zero-mean noise, the estimate is unbiased

$$\mathbb{E}\left[\hat{G}^{(p)}(\mathrm{e}^{j\omega_k})\right] = G^{(p)}(\mathrm{e}^{j\omega_k}) + \mathbb{E}\left[\tilde{V}^{(p)}(\mathrm{e}^{j\omega_k})\right]\tilde{U}^\dagger(\mathrm{e}^{j\omega_k}) = G^{(p)}(\mathrm{e}^{j\omega_k}). \tag{6.37}$$

The covariance of the estimate is given by

$$\begin{aligned}\mathbb{E}\Bigg[\left(\hat{G}^{(p)}(\mathrm{e}^{j\omega_k}) - G^{(p)}(\mathrm{e}^{j\omega_k})\right)^{\mathsf{H}} \left(\hat{G}^{(p)}(\mathrm{e}^{j\omega_m}) - G^{(p)}(\mathrm{e}^{j\omega_m})\right)\Bigg] \\ = \left(\tilde{U}^\dagger(\mathrm{e}^{j\omega_k})\right)^{\mathsf{H}} \mathbb{E}\left[\left(\tilde{V}^{(p)}(\mathrm{e}^{j\omega_k})\right)^{\mathsf{H}} \tilde{V}^{(p)}(\mathrm{e}^{j\omega_m})\right] \tilde{U}^\dagger(\mathrm{e}^{j\omega_m}).\end{aligned} \tag{6.38}$$

From Section 6.3 of Ljung (1999) and the independence across different experiments, we have

$$\mathbb{E}\left[\left(\tilde{V}^{(p)}(\mathrm{e}^{j\omega_k})\right)^{\mathsf{H}} \tilde{V}^{(p)}(\mathrm{e}^{j\omega_m})\right] = \begin{cases} \left(\Phi_{v_p}(\mathrm{e}^{j\omega_k}) + \rho_p(N_p)\right)\mathbb{I}, & \omega_k = \omega_m \\ \mathbb{0}, & \omega_k \neq \omega_m \end{cases}. \tag{6.39}$$

Substituting (6.39) into (6.38) completes the proof. □

### 6.3.2 Order-Revealing Decomposition for LTP Systems

With the frequency response of the lifted system estimated, the order-revealing decomposition analogous to (6.28) can be developed for LTP systems.

Take the IDFT of the $(l,m)$-th block element of $\hat{G}(\mathrm{e}^{j\omega_k})$ in (6.31),

$$w_{l,m}(n) := \frac{1}{N_p}\sum_{k=0}^{N_p-1} \hat{G}_{l,m}(\mathrm{e}^{j\omega_k}) \exp\left(j\frac{2\pi nk}{N_p}\right) = \frac{1}{N_p}\sum_{k=0}^{N_p-1}\sum_{s=0}^{\infty} g^l_{sT^\star+l-m} \exp\left(-j\frac{2\pi(s-n)k}{N_p}\right). \tag{6.40}$$

Since the summation over $k$ is on the whole unit circle, it is only non-zero when $s-n = iN_p$, $i \in \mathbb{Z}$. We have

$$w_{l,m}(n) = \begin{cases} \sum_{i=0}^{\infty} g^l_{(iN_p+n)T^\star+l-m}, & nT^\star + l - m > 0, \\ \sum_{i=0}^{\infty} g^l_{(iN_p+N_p+n)T^\star+l-m}, & nT^\star + l - m \leq 0. \end{cases} \tag{6.41}$$

Define the time-aliased periodic impulse response as

$$h_r^t := \sum_{i=0}^{\infty} g_{r+iN_pT^\star}^t, \quad r = 1,2,\ldots,N_pT^\star. \tag{6.42}$$

According to the definition of $g_r^t$ (6.2), $\forall p = 0,1,\ldots,r-1$,

$$h_r^t = C_t A_{t-1}\ldots A_{t-p}\left(\mathbb{I} - \Psi_{A,(t-p)}^{N_p}\right)^{-1} A_{t-p-1}\ldots A_{t-r+1}B_{t-r}. \tag{6.43}$$

Therefore, the periodic block Hankel matrix of $h_r^t$ can be decomposed as follows

$$H_p^\tau := \begin{bmatrix} h_1^\tau & h_2^\tau & \cdots & h_r^\tau \\ h_2^{\tau+1} & h_3^{\tau+1} & \cdots & h_{r+1}^{\tau+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_q^{\tau+T^\star-1} & h_{q+1}^{\tau+T^\star-1} & \cdots & h_{q+r-1}^{\tau+T^\star-1} \end{bmatrix} = \mathcal{O}_q^\tau \left(\mathbb{I} - \Psi_{A,\tau}^{N_p}\right)^{-1} \mathcal{C}_r^\tau, \tag{6.44}$$

where $q+r-1 \leq N_pT^\star$. By selecting $q,r$ such that $qn_y, rn_u \geq n_x$, we have

$$\operatorname{rank}\left(H_p^\tau\right) = \operatorname{rank}\left(\mathcal{O}_q^\tau\right) = \operatorname{rank}\left(\left(\mathbb{I} - \Psi_{A,\tau}^{N_p}\right)^{-1}\right) = \operatorname{rank}\left(\mathcal{C}_r^\tau\right) = n_x. \tag{6.45}$$

Note that the rank requirements on $q$ and $r$ put a lower bound on $N_p$. Then the range space of $H_p^\tau$ coincides with that of $\mathcal{O}_q^\tau$. Thus, $\mathcal{O}_q^\tau$ can be identified, up to a similarity transform, by performing SVD on $H_p^\tau$. From the extended observability matrix, the matrices $\bar{A}$ and $\bar{C}$ can be estimated by the same shifting method as in the time-domain subspace identification of LTP systems (Verhaegen and Yu, 1995). The input matrix $\bar{B}$ can be estimated by least-squares fit to the time-aliased impulse response.

### 6.3.3 Algorithm & Consistency Analysis

Built on the decomposition (6.44), we propose Algorithm 6.2 for frequency-domain subspace identification of LTP systems with periodic inputs.

The computational complexity of Algorithm 6.2 is dominated by solving the least squares problem (6.47), which has a complexity of $O\left(n_x^2 \cdot n_u^2 \cdot N_p \cdot T^{\star 2}\right)$. The consistency of Algorithm 6.2 is shown in Theorem 6.1.

**Theorem 6.1.** *Let $A_t$, $B_t$, and $C_t$ define the minimal LTP state-space model (6.1). Let $\hat{A}_t$, $\hat{B}_t$, and $\hat{C}_t$ be the estimated state matrices by Algorithm 6.2. Under Assumption 6.1, there exist non-singular periodic matrices $T_t \in \mathbb{R}^{n_x \times n_x}$, $T_t = T_{t+T^\star}$ such that, w.p. 1,*

$$\lim_{N_p\to\infty} \left\| \begin{bmatrix} A_t & B_t \\ C_t & \mathbb{0} \end{bmatrix} - \begin{bmatrix} T_{t+1} & \mathbb{0} \\ \mathbb{0} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \hat{A}_t & \hat{B}_t \\ \hat{C}_t & \mathbb{0} \end{bmatrix} \begin{bmatrix} T_t^{-1} & \mathbb{0} \\ \mathbb{0} & \mathbb{I} \end{bmatrix} \right\|_F = 0, \tag{6.50}$$

*for a fixed choice of $q,r$.*

**Algorithm 6.2** Frequency-domain subspace identification of LTP systems with periodic inputs

1: Lift the input-output data $u_t^i$, $y_t^i$ to $\tilde{u}_k^i$, $\tilde{y}_k^i$ for $k = 0, \ldots, N_p - 1$ and $i = 1, \ldots, J$, as in (6.5).
2: Estimate the frequency response $\hat{G}(\mathrm{e}^{j\omega_k})$ of the lifted system from $\tilde{u}_k^i$, $\tilde{y}_k^i$ by (6.30) and (6.31).
3: Apply the IDFT on each block element $\hat{G}_{l,m}(\mathrm{e}^{j\omega_k})$ of $\hat{G}(\mathrm{e}^{j\omega_k})$ according to (6.40) and denote it as $\hat{w}_{l,m}(n)$.
4: Construct the time-aliased periodic impulse response $\left\{\hat{h}_r^t\right\}$ for $r = 1, \ldots, N_p T^\star$ and $t = 0, \ldots, T^\star - 1$ by rearranging elements in $\hat{w}_{l,m}(n)$ according to (6.41) and (6.42).
5: Construct $\hat{H}_p^\tau$ for $\tau = 0, \ldots, T^\star - 1$, according to (6.44).
6: Calculate the SVD of $\hat{H}_p^\tau$ for $\tau = 0, \ldots, T^\star - 1$: $\hat{H}_p^\tau = \hat{U}_\tau \hat{\Sigma}_\tau \hat{V}_\tau^\top$.
7: Determine a model order $n_x$ and define $\hat{U}_\tau := \left[\hat{U}_\tau^s \; \hat{U}_\tau^o\right]$, where $\hat{U}_\tau^s \in \mathbb{R}^{qn_y \times n_x}$.
8: The estimated state-space model is given as

$$\hat{A}_\tau = (J_1 \hat{U}_{\tau+1}^s)^\dagger J_2 \hat{U}_\tau^s, \quad \hat{C}_\tau = J_3 \hat{U}_\tau^s, \quad \tau = 0, \ldots, T^\star - 1, \tag{6.46}$$

$$\hat{\bar{B}} = \underset{\bar{B}}{\operatorname{argmin}} \sum_{r=1}^{N_p T^\star} \sum_{\tau=0}^{T^\star - 1} \left\| \hat{h}_r^\tau - \hat{Q}_r^\tau B_{\tau - r} \right\|_F^2, \tag{6.47}$$

where

$$J_1 := \left[\mathbb{I} \quad \mathbb{0}_{(q-1)n_y \times n_y}\right], \quad J_2 := \left[\mathbb{0}_{(q-1)n_y \times n_y} \quad \mathbb{I}\right], \quad J_3 := \left[\mathbb{I} \quad \mathbb{0}_{n_y \times (q-1)n_y}\right], \tag{6.48}$$

$$\hat{U}_{T^\star}^s = \hat{U}_0^s, \quad \hat{Q}_r^\tau := \hat{C}_\tau \left(\mathbb{I} - \Psi_{\hat{A},\tau}^{N_p}\right)^{-1} \hat{A}_{\tau-1} \ldots \hat{A}_{\tau-r+1}. \tag{6.49}$$

*Proof.* Let $\Delta G(\mathrm{e}^{j\omega_k}) := \hat{G}(\mathrm{e}^{j\omega_k}) - G(\mathrm{e}^{j\omega_k})$, $\Delta w_{l,m}(n) := \hat{w}_{l,m}(n) - w_{l,m}(n)$. We have

$$\Delta w_{l,m}(n) = \frac{1}{N_p} \sum_{k=0}^{N_p - 1} \Delta G_{l,m}(\mathrm{e}^{j\omega_k}) \exp\left(j \frac{2\pi n k}{N_p}\right), \tag{6.51}$$

which can be seen as the sample mean of zero-mean independent random variables (McKelvey et al., 1996). From Lemma 6.1, we know that the covariances of the random variables are bounded. Thus, according to the law of large numbers,

$$\lim_{N_p \to \infty} \Delta w_{l,m}(n) = 0, \quad \text{w.p.1}, \tag{6.52}$$

Then let $\Delta h_r^t := \hat{h}_r^t - h_r^t$, $\Delta H_p^\tau := \hat{H}_p^\tau - H_p^\tau$. We have $\lim_{N_p \to \infty} \Delta h_r^t = 0$, w.p. 1, which implies that, for $\tau = 0, 1, \ldots, T^\star - 1$,

$$\lim_{N_p \to \infty} \left\| \Delta H_p^\tau \right\|_F = 0, \quad \text{w.p.1}. \tag{6.53}$$

Let $\left\| \Delta H_p^\tau \right\|_F \leq \varepsilon$. According to the proof of Lemma 4 in McKelvey et al. (1996), there exist a matrix $P_\tau$ satisfying $\|P_\tau\|_F \leq 4\varepsilon / \sigma_{n_x}(H_p^\tau)$ and a non-singular matrix $T_\tau$ such that

$$\hat{U}_\tau^s = (U_\tau^s + U_\tau^o P_\tau) T_\tau, \tag{6.54}$$

where $H_p^\tau = [U_\tau^s \; U_\tau^o]\Sigma_\tau V_\tau^\top$ is the SVD of $H_p^\tau$. Then, we have

$$T_{\tau+1}\hat{A}_\tau T_\tau^{-1} = \left(J_1(U_{\tau+1}^s + U_{\tau+1}^o P_{\tau+1})\right)^\dagger J_2(U_\tau^s + U_\tau^o P_\tau), \quad \hat{C}_\tau T_\tau^{-1} = J_3(U_\tau^s + U_\tau^o P_\tau). \tag{6.55}$$

Note that $J_1 U_{\tau+1}^s A_\tau = J_2 U_\tau^s$ and $C_\tau = J_3 U_\tau^s$. Then, from Theorem 5.3.1 in Golub and Van Loan (2012) on the sensitivity of the least squares estimate, for a sufficiently small $\varepsilon$ such that the regressor does not lose rank, there exists constants $c_\tau, c'_\tau$, such that

$$\left\|T_{\tau+1}\hat{A}_\tau T_\tau^{-1} - A_\tau\right\|_F \le c_\tau \varepsilon, \quad \left\|\hat{C}_\tau T_\tau^{-1} - C_\tau\right\|_F \le c'_\tau \varepsilon. \tag{6.56}$$

For the estimate $\hat{\bar{B}}$ in (6.47), let

$$Q_r^\tau := C_\tau \left(\mathbb{I} - \Psi_{A,\tau}^M\right)^{-1} A_{\tau-1} \ldots A_{\tau-r+1}. \tag{6.57}$$

Then, a simple calculation shows that

$$\left\|\hat{Q}_r^\tau T_{\tau-r+1}^{-1} - Q_r^\tau\right\|_F = O(\varepsilon). \tag{6.58}$$

Since $\Delta h_r^t = O(\varepsilon)$, again from Theorem 5.3.1 in Golub and Van Loan (2012), for a sufficiently small $\varepsilon$,

$$\left\|T_{\tau-r+1}\hat{B}_{\tau-r} - B_{\tau-r}\right\|_F = O(\varepsilon). \tag{6.59}$$

The above equation, together with (6.53) and (6.56), completes the proof. □

### 6.3.4 Numerical Results

This subsection tests the proposed algorithm against multiple time-domain subspace identification algorithms for LTP systems with two numerical examples. Example 1 is based on the flapping dynamics of wind turbines, which is taken from Felici et al. (2007). The true dynamics of the system are given by

$$\left[\begin{array}{c|c} A_0 & B_0 \\ \hline C_0 & 0 \end{array}\right] = \left[\begin{array}{cc|c} 0 & 0.0734 & -0.07221 \\ -6.5229 & -0.4997 & -9.6277 \\ \hline 1 & 0 & 0 \end{array}\right], \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & 0 \end{array}\right] = \left[\begin{array}{cc|c} -0.0021 & 0 & 0 \\ -0.0138 & 0.5196 & 0 \\ \hline 0 & 0 & 0 \end{array}\right],$$

where $n_x = 2$, $n_y = n_u = 1$, $T^\star = 2$. Example 2 is used in Hench (1995) with the dynamics

$$\left[\begin{array}{c|c} A_0 & B_0 \\ \hline C_0 & 0 \end{array}\right] = \left[\begin{array}{cc|c} 1 & 1 & 0 \\ 0 & 2 & 1 \\ \hline 1 & 0 & 0 \end{array}\right], \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & 0 \end{array}\right] = \left[\begin{array}{cc|c} \frac{1}{5} & 1 & 0 \\ 0 & \frac{2}{5} & 1 \\ \hline 2 & 0 & 0 \end{array}\right], \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & 0 \end{array}\right] = \left[\begin{array}{cc|c} 3 & 1 & 1 \\ 0 & 1 & 2 \\ \hline 1 & 1 & 0 \end{array}\right],$$

where $n_x = 2$, $n_y = n_u = 1$, $T^\star = 3$. Both systems are then normalized to have an average steady-state gain of 1.

The compared algorithms are: 1) Algorithm 6.2 in this paper (*Freq*), 2) the MOESP algorithm in

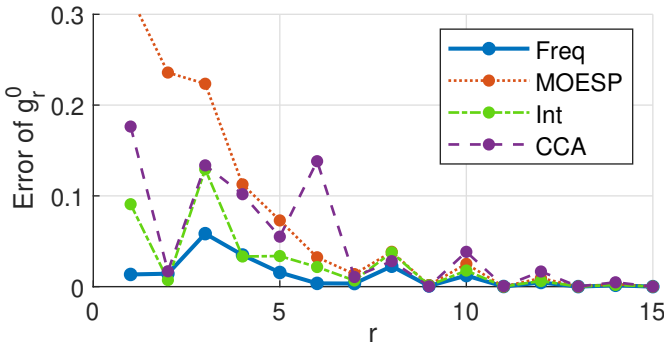



Figure 6.6: Errors in the periodic impulse response estimation for example 1.

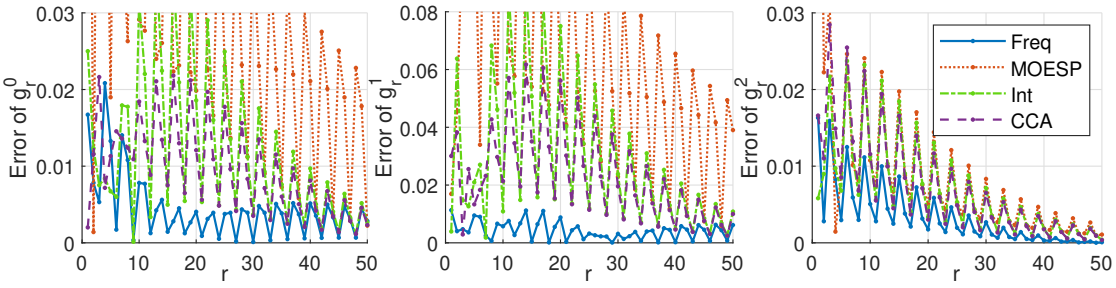



Figure 6.7: Errors in the periodic impulse response estimation for example 2.

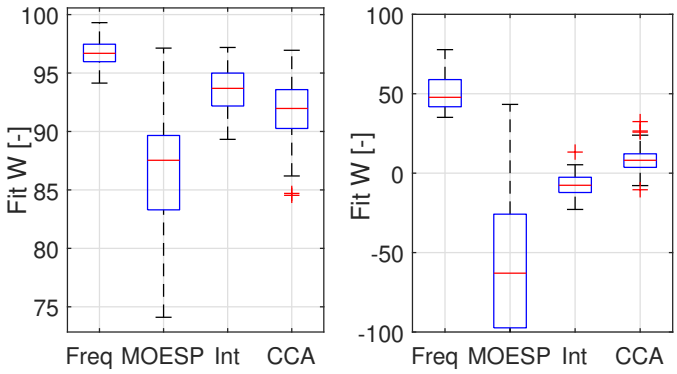
Verhaegen and Yu (1995) (*MOESP*), 3) the intersection algorithm in Hench (1995) (*Int*), and 4) the CCA algorithm in Lemma 9.2 of Cox (2018) specialized for LTP systems (*CCA*).

In both examples, the following simulation configuration and parameters are used. For each input-output data sequence, the systems are excited by periodic input of i.i.d. unit Gaussian entries $u_t \sim \mathcal{N}(0,1)$ from zero initial conditions. The outputs are contaminated with i.i.d. unit Gaussian noise $v(t) \sim \mathcal{N}(0,1)$. The identification data are collected with $N_p = 50$, $J = 10 \cdot T^\star$ after the transient effect becomes negligible. The number of block rows $q$ for the Hankel matrices in all methods is selected by cross-validation. The model order $n_x$ is assumed to be known.

The identification results are shown in Figures 6.6 and 6.7 for examples 1 and 2, respectively, in terms of the absolute estimation errors of the periodic impulse responses $g_r^\tau$, as the state-space matrices depend on unknown similarity transforms. In example 1, the system is autonomous at $\tau = 1$, so only the impulse responses at $\tau = 0$ are shown. As seen from both figures, the estimation error of the proposed method is smaller than that of the other three time-domain methods. In particular, for example 2, the time-domain methods fail to provide a meaningful estimation of the system, whereas the proposed frequency-domain method can still obtain reasonable results.

To quantitatively assess the performance of the identification schemes, 100 Monte Carlo simulations with different noise realizations were conducted for both examples using the same performance metric $W$ as in Section 6.2.3. The boxplots for both examples are shown in Fig-

ure 6.8. In both examples, the proposed method has a better fitting performance compared to the time-domain methods.




(a) Example 1 (b) Example 2

Figure 6.8: Comparison of fitting performance with Monte Carlo simulations.

The above results demonstrate that the proposed method performs better than the time-domain methods when periodic input-output data are available. This advantage is mainly because it makes use of the periodic nature of the identification data. This gives the complete input history of the system or, in other words, the initial condition, whereas in the time-domain method, past inputs are assumed unknown.

Finally, we demonstrate the consistency property proved in Theorem 6.1 by conducting Monte Carlo simulations of example 1 with increasing data length $N_p$. The results are shown in Figure 6.9, where the estimation error is characterized by the MSE of the periodic impulse response estimate. It can be seen that the estimate is consistent with a convergence rate of $1/N_p$.

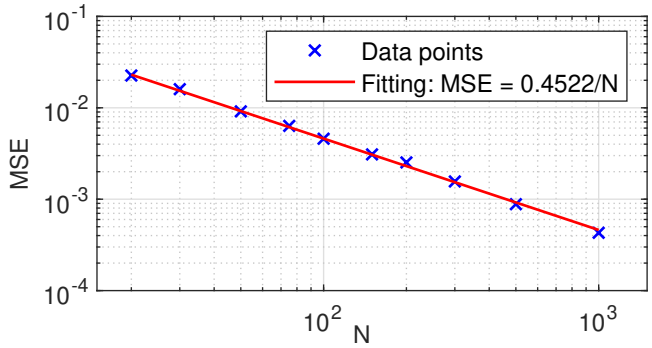



Figure 6.9: MSE of the frequency-domain subspace estimate under different data lengths.

## 6.4 Summary

This chapter presents two methods for identifying linear time-periodic (LTP) systems. The first method uses grouped atomic norm regularization on linear time-invariant (LTI) sub-models from

the switching reformulation. An essential requirement for the identification to be successful is that the sub-models should have the same pole locations. Therefore, the atomic norm regularizer for LTI systems is extended to LTP systems with the group lasso technique to impose this additional structural constraint. This method obtains uniform low-order models of LTP systems, and simulations show it has a better model fit than existing methods under high noise levels. The main message of this work is that the LTP system identification problem cannot be fully tackled by LTI system theory. The key to enhancing the performance of LTP system identification is incorporating specific structural constraints arising from periodicity with appropriate frameworks.

The second method uses periodic identification data to extend the frequency-domain subspace identification algorithm to the lifted LTP system. This method applies a two-step approach: the generalized ETFE of the lifted system is first obtained from the identification data. The time-aliased periodic impulse response derived from the lifted frequency response is used to construct an order-revealing decomposition of the original LTP system, from which the general framework of subspace identification can be utilized. The proposed algorithm complements the available subspace identification algorithms for LTP systems and shows an advantage in model fitting from numerical simulation when periodic data are available.

# 7 Identification of Limit Cycle Dynamics with Periodic Models

This chapter identifies limit cycle dynamics in nonlinear systems with linear periodic models. To obtain a local model close to the limit cycle, direct linearization around the limit cycle is conducted in Allen and Sracic (2009). However, this approach fails to capture the dynamics along the limit cycle, i.e., the velocity at which the perturbed trajectories traverse the points on the limit cycle while converging to it. In this work, the dynamics are first decomposed onto the so-called *transverse coordinates* (Manchester, 2011). Next, the dynamics around the limit cycle are modeled as a periodic system parametrized with the location on the limit cycle. The system can be approximated near the limit cycle with a locally linearized model known as the linear periodically parameter-varying (LPPV) model.

This approach translates the limit cycle identification problem into a periodic function learning problem of system matrices, which is often solved by basis function decomposition onto a higher dimensional nonlinear feature space. The kernel method allows this mapping to be done implicitly by specifying the corresponding infinite-dimensional Hibert space with a *kernel function*. The system matrices can then be estimated in this function space with ridge regularization. Such methods have been previously used for nonparametric identification of LPV systems in Laurain et al. (2012) with an input-output model and Rizvi et al. (2018) with a state-space model. This chapter extends the method proposed in Rizvi et al. (2018) with a separate kernel design for each element of the system matrices, and the periodicity in the learned system matrices is enforced via periodic kernel design. In addition, the flexibility of kernel design makes it possible to include additional system parameters in the model by augmenting the periodic kernel with standard non-periodic kernels.

The algorithm is first tested on the Van der Pol oscillator. The identified model is demonstrated to be close to analytical linearization when training data are close to the limit cycle and outperforms analytical linearization in terms of prediction accuracy when the training data are close to the prediction task. Then, the algorithm is applied to a simplified kinematic model of a tethered kite controlled to fly along a periodic figure-of-eight trajectory for airborne wind energy generation (Ahrens et al., 2013). Accurate predictions can be obtained with an additional system parameter.

The proposed method performs significantly better than global nonlinear identification without knowledge of the limit cycle.

## 7.1 Transverse Dynamics of Limit Cycles

In this section, the background of the limit cycle and its transverse dynamics are summarized. See Hale (1980); Manchester (2011) for detailed definitions and derivations.

Consider a nonlinear system described by a set of ordinary differential equations (ODE's):

$$\dot{x} = f(x,d), \tag{7.1}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector and $d \in \mathbb{R}^{n_d}$ is the exogenous input. The autonomous solution of this system, i.e., $\dot{x} = f(x,0)$, starting from an initial condition $x(0) = x_0$ is denoted by $x(t) =: \Phi_f(x_0,t)$. The system exhibits limit cycle behaviour if it has a $T^\star$-periodic solution $x^\star(t) = \Phi_f(x_0^\star,t)$, i.e., $T^\star > 0$ is the minimum period such that the relationship $x^\star(t) = x^\star(t+T^\star)$ holds for all $t$. Then, the limit cycle is defined as $\Gamma_f^\star := \{x \in \mathbb{R}^{n_x} : x = x^\star(\tau) | \tau \in [0,T^\star)\}$, where $\tau \in [0,T^\star)$ parametrizes the location on the limit cycle. In this study, we consider asymptotically stable periodic orbits. The periodic orbit $\Gamma_f^\star$ is said to be asymptotically stable if it fulfills Lyapunov stability, i.e., $\forall \varepsilon > 0$, $\exists \delta > 0$ such that $\forall x_0 \in \mathbb{R}^{n_x}$ with $\mathrm{dist}(x_0,\Gamma_f^\star) < \delta$, we have $\mathrm{dist}\left(\Phi_f(x_0,t),\Gamma_f^\star\right) < \varepsilon$, $\forall t > 0$ and $\lim_{t\to\infty} \mathrm{dist}\left(\Phi_f(x_0,t),\Gamma_f^\star\right) = 0$, where $\mathrm{dist}(x,\Gamma_f^\star) := \inf_{y\in\Gamma_f^\star} ||y-x||_2$ defines the distance from a point to the orbit.

At each $\tau$, an $(n_x-1)$-dimensional hyperplane $S(\tau)$ that is transversal to $\Gamma_f^\star$ can be constructed, i.e., $\dot{x}^\star(\tau) \notin S(\tau)$. The transversal hyperplanes are uniquely defined by normal vectors denoted by $z(\tau)$. The transversality condition can be rewritten in terms of the normal vector as $z(\tau)^\top \dot{x}^\star(\tau) > 0$, $\forall \tau \in [0,T^\star)$. On this hyperplane, a new coordinate system is defined such that the origin is $x^\star(\tau)$, and the coordinate axes can be chosen as any orthonormal basis that spans the surface $S(\tau)$. The coordinates of a given state $x \in S(\tau)$ in this new coordinate frame are denoted by $x_\perp \in \mathbb{R}^{n_\perp}$, where $n_\perp := n_x - 1$. Thus, a mapping of the state to its transverse coordinates is created for a given family of transversal surfaces moving along the periodic orbit: $x \to (x_\perp,\tau)$. The collection of the basis vectors of $S(\tau)$ defines a projection operator $\Pi(\tau) := [\xi_1(\tau)\ \dots\ \xi_{n_\perp}(\tau)]^\top \in \mathbb{R}^{n_\perp \times n_x}$ that characterizes the transformation to the transverse coordinates:

$$x_\perp = \Pi(\tau)(x - x^\star(\tau)), \tag{7.2}$$

and the inverse relationship is

$$x = x^\star(\tau) + \Pi(\tau)^\top x_\perp, \tag{7.3}$$

since $\Pi(\tau)\Pi(\tau)^\top = \mathbb{I}$ due to the orthonormality of the basis vectors.

The most straightforward choice of these hyperplanes $S(\tau)$ is then those that are orthogonal to

the orbit, i.e., the normal vectors are set to be tangential to the flow as

$$z^{\text{orth}}(\tau) := \frac{\dot{x}^\star(\tau)}{||\dot{x}^\star(\tau)||_2}. \tag{7.4}$$

However, this choice leads to singularities, especially around $\tau$ sections where the curvature of the orbit is large (Manchester, 2011). These singularities violate the so-called well-posedness condition that arises from the nonlinear $\tau$ dynamics. This condition restricts the region where the transformation to transverse coordinates is well-defined. An alternative set of surfaces is considered, originally proposed in Ahbe et al. (2022). These surfaces, referred to as center surfaces, connect $x^\star(\tau)$ with a fixed center (e.g., the geometric center of the limit cycle) with the first basis vector $\xi_1(\tau)$ being

$$\xi_1^{\text{center}}(\tau) := \frac{x^\star(\tau) - x_c}{||x^\star(\tau) - x_c||_2}, \tag{7.5}$$

where $x_c$ represents the designated center point. The normal vector $z^{\text{center}}(\tau)$ can be consequently determined by minimizing the angle between the center surface and the orthogonal surface, i.e., $z^{\text{center}}(\tau)$ is selected as the projection of $z^{\text{orth}}(\tau)$ onto the plane that is normal to $\xi_1^{\text{center}}(\tau)$.

The corresponding hyperplane must first be determined to convert a state $x$ to its transverse counterpart $(x_\perp, \tau)$. The problem can be reformulated as finding the $\tau$ that satisfies the hyperplane equation and minimizes the distance between $x$ and the corresponding point on the limit cycle:

$$\begin{aligned} \min_{\tau} \quad & ||x - x^\star(\tau)||_2, \\ \text{s.t.} \quad & z(\tau)^\top (x - x^\star(\tau)) = 0. \end{aligned} \tag{7.6}$$

Using the transformations established in (7.2) and (7.3), when the nonlinear model (7.1) is known, the dynamics of the transverse states can be analytically obtained (Manchester, 2011):

$$\dot{x}_\perp = \dot{\tau}\left[\frac{d}{d\tau}\Pi(\tau)\right]\Pi(\tau)^\top x_\perp + \Pi(\tau) f(x^\star(\tau) + \Pi(\tau)^\top x_\perp) - \Pi(\tau) f(x^\star(\tau))\dot{\tau}, \tag{7.7a}$$

$$\dot{\tau} = \frac{z(\tau)^\top f(x^\star(\tau) + \Pi(\tau)^\top x_\perp)}{z(\tau)^\top f(x^\star(\tau)) - \frac{dz(\tau)}{d\tau}^\top \Pi(\tau)^\top x_\perp}, \tag{7.7b}$$

where the aforementioned well-posedness condition is given by

$$z(\tau)^\top f(x^\star(\tau)) - \frac{dz(\tau)}{d\tau}^\top \Pi(\tau)^\top x_\perp \neq 0. \tag{7.8}$$

The transverse dynamics can be further linearized at $x_\perp = 0$ and lead to an affine model in the following form:

$$\dot{x}_\perp = A(\tau)x_\perp + B(\tau)d, \tag{7.9a}$$

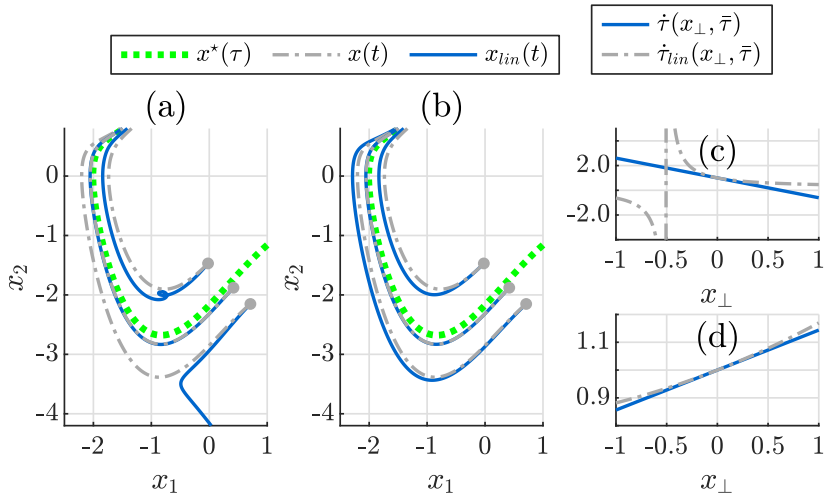



Figure 7.1: Effects of transversal surface selection. (a),(b): trajectory simulations, (c),(d): $\tau$ dynamics at a sharp turn, (a),(c): orthogonal transversal surfaces, (b),(d): center transversal surfaces.

$$\dot{\tau} = 1 + g(\tau)x_\perp + h(\tau)d, \tag{7.9b}$$

where $A(\tau) : [0, T^\star) \to \mathbb{R}^{n_\perp \times n_\perp}$, $B(\tau) : [0, T^\star) \to \mathbb{R}^{n_\perp \times n_d}$, $g(\tau) : [0, T^\star) \to \mathbb{R}^{1 \times n_\perp}$, and $h(\tau) : [0, T^\star) \to \mathbb{R}^{1 \times n_d}$ are periodically-varying matrix functions of $\tau$.

**Example 7.1.** *(Comparison of transversal hyperplane design) Consider the nonlinear benchmark system known as the Van der Pol oscillator, described by:*

$$\dot{x}_1 = x_2, \tag{7.10a}$$
$$\dot{x}_2 = \mu(1 - x_1^2)x_2 - x_1 + D\sin(\omega t), \tag{7.10b}$$

*where a sinusoidal forcing term is added, corresponding to the external input $d$ in* (7.9)*. It is well-known that this nonlinear system has a stable limit cycle. The damping coefficient $\mu$ is set to 1, which results in a limit cycle with period $T^\star = 6.663$.*

*In Figure 7.1, nonlinear trajectories generated from* (7.10) *with $D = 0$, denoted by $x(t)$, are compared to those obtained from the analytical transverse linear approximation $x_{lin}(t)$ using (a) orthogonal, and (b) center surfaces (the center point $x_c$ is chosen as the origin). For orthogonal surfaces, the well-posedness condition* (7.8) *is violated around the sharp turn where the surfaces clash into each other, which causes a discontinuity in the $\tau$ dynamics (Figure 7.1(c)). Around these regions, the transverse linear dynamics become unstable for large $x_\perp$ values (Figure 7.1(a)). This undesired behavior is prevented by using center surfaces, where the linear dynamics $\dot{\tau}_{lin}$ can effectively approximate $\dot{\tau}$ (Figure 7.1(d)). These observations prompt the use of center surfaces for approximating local limit cycle dynamics with LPPV models.*

In this work, the nonlinear dynamics (7.1) is unknown, and we are interested in identifying the

locally linearized model of the transverse system (7.9) by collecting training data of the original state trajectories, their time derivatives, and the exogenous inputs, namely $\{x(t_k), \dot{x}(t_k), d(t_k)\}_{k=1}^{N}$. It is also assumed that the periodic orbit $\Gamma_f^\star$ is known. This becomes a function learning problem of periodic matrix functions $A(\tau), B(\tau), g(\tau), h(\tau)$.

**Remark 7.1.** *Note that if $x$ is on the limit cycle, i.e., $x_\perp = 0$, $\tau$ would be equal to $t$ when no exogenous input is applied. Otherwise, the $\tau$ dynamics would differ from $t$, and the transverse model encapsulates this behavior. The LTV approach in Allen and Sracic (2009) adopts the following LTV model which ignores the $\tau$ dynamics* (7.9b)*: $\dot{\tilde{x}} = \tilde{A}(t)\tilde{x} + \tilde{B}(t)d$, where $\tilde{x}(t) := x(t) - x^\star(t)$. This can lead to a large discrepancy in trajectories when the magnitude of $x_\perp$ is large.*

## 7.2 Identification of Linear Periodically Parameter-Varying Models

This section introduces a kernel-based method for identifying the linearized transverse model of the limit cycle dynamics (7.9).

The dynamics (7.9) can then be compactly rewritten as:

$$\zeta = \Omega(\tau)\theta, \tag{7.11}$$

where $\theta := \begin{bmatrix} x_\perp^\top & d^\top \end{bmatrix}^\top \in \mathbb{R}^{n_\theta}$, where $n_\theta := n_\perp + n_d$, $\zeta := \begin{bmatrix} \dot{x}_\perp^\top & \dot{\tau} - 1 \end{bmatrix}^\top \in \mathbb{R}^{n_x}$, and

$$\Omega(\tau) := \begin{bmatrix} A(\tau) & B(\tau) \\ g(\tau) & h(\tau) \end{bmatrix} : [0, T^\star) \to \mathbb{R}^{n_x \times n_\theta}. \tag{7.12}$$

To obtain $\theta(t_k)$, $\zeta(t_k)$, and $\tau(t_k)$ from the collected trajectory data, problem (7.6) is solved for each $\tau(t_k)$ by a nonlinear solver initialized from $\tau(t_{k-1})$. The transverse coordinates $x_\perp(t_k)$ are then computed using the projection in (7.2). Finally, the time derivatives of the transverse states $(\dot{x}_\perp(t_k), \dot{\tau}(t_k))$ can be calculated from $\dot{x}(t_k)$ using the nonlinear analytical expressions from Theorem 1 in Manchester (2011). Thus, the transformed dataset $\{\theta(t_k), \zeta(t_k), \tau(t_k)\}_{k=1}^{N}$ is obtained.

### 7.2.1 Kernel-Based Identification

In this subsection, we approach the function learning problem using the basis decomposition interpretation introduced in Section 3.1. Assume that the underlying function can be decomposed into a set of basis functions:

$$\Omega_i(\tau) = \sum_{m=1}^{n_\psi} w_m^i \bar{\psi}_m^i(\tau) = W_i \bar{\Psi}_i(\tau), \tag{7.13}$$

where $\Omega_i(\tau)$ denotes the $i$-th row of $\Omega(\tau)$, $\bar{\psi}^i_m(\tau) : [0, T^\star) \to \mathbb{R}^{1\times n_\theta}$ represent the preselected vector-valued basis functions, $w^i_m \in \mathbb{R}$ are the associated weights, and

$$\bar{\Psi}_i(\tau) := \operatorname{col}\left(\bar{\psi}^i_1(\tau), \ldots, \bar{\psi}^i_{n_\psi}(\tau)\right), \quad W_i := \begin{bmatrix} w^i_1 & \ldots & w^i_{n_\psi} \end{bmatrix} \tag{7.14}$$

collect the basis functions and the weights, respectively. The matrix $\bar{\Psi}_i(\tau)$ is also known as the *feature map* in machine learning literature. Each row $\Omega_i(\tau)$ of the system matrix is considered separately and solved independently.

The learning problem is then posed as a regularized least-squares problem:

$$\min_{W_i} \quad \sum_{k=1}^{N} \left(\zeta_i(t_k) - W_i \bar{\Psi}_i(\tau(t_k))\theta(t_k)\right)^2 + \lambda_i ||W_i||_2^2, \tag{7.15}$$

where a ridge regularization term is applied with a weight of $\lambda_i \in \mathbb{R}$. The predictions of state derivatives $\zeta_i$ is denoted as

$$\hat{\zeta}_i(t_k) := W_i \bar{\Psi}_i(\tau(t_k))\theta(t_k). \tag{7.16}$$

Problem (7.15) can be solved directly. However, selecting the basis functions $\bar{\Psi}_i(\tau)$ is not trivial, and the dimension $n_\psi$ is typically very large to achieve a good prediction accuracy. Instead, the kernel method is used to reformulate the problem. In detail, by formulating the dual problem of (7.15), it has been shown that the optimal solution of the weights $W_i$ lies in the span of the training data (Rizvi et al., 2018; Tóth et al., 2011):

$$W_i = \sum_{k=1}^{N} \alpha_{i,k} \theta(t_k)^\top \bar{\Psi}_i(\tau(t_k))^\top, \tag{7.17}$$

where $\alpha_{i,k} \in \mathbb{R}$ are the coefficients associated with each training point. The predicted $\zeta_i$ can thus be expressed as

$$\hat{\zeta}_i(t_{k'}) = \sum_{k=1}^{N} \alpha_{i,k} \theta(t_k)^\top \bar{\Psi}_i(\tau(t_k))^\top \bar{\Psi}_i(\tau(t_{k'}))\theta(t_{k'}). \tag{7.18}$$

Then, problem (7.15) can be reformulated in terms of $\alpha_i := [\alpha_{i,1} \ \alpha_{i,2} \ \ldots \ \alpha_{i,N}]^\top$, which only depends on the inner product of the feature map $\bar{K}_i(\tau, \tau') := \bar{\Psi}_i(\tau)^\top \bar{\Psi}_i(\tau') \in [0, T^*) \times [0, T^*) \to \mathbb{R}^{n_\theta \times n_\theta}$ instead of $\bar{\Psi}_i(\tau)$ itself. This inner product function $\bar{K}_i(\cdot, \cdot)$ is known as the kernel function. Since $n_\psi$ is usually much larger than $n_\theta$, it is often easier to directly design $\bar{K}_i$ instead of $\bar{\Psi}_i$ to avoid explicitly choosing the map while still implicitly working with features of higher or infinite dimensions. The idea of replacing the inner product of the feature map with the kernel function is known as the *kernel trick* (Schölkopf, 2001). Substituting the kernel into (7.18), we obtain

$$\hat{\zeta}_i(t_{k'}) = \sum_{k=1}^{N} \alpha_{i,k} \theta(t_k)^\top \bar{K}_i(\tau(t_k), \tau(t_{k'}))\theta(t_{k'}). \tag{7.19}$$

Assuming that the elements of the system matrices can be modeled independently from each other, the kernel functions $\bar{K}_i$ are designed as diagonal matrices, i.e., $\bar{K}_i = \operatorname{diag}(k_{i,1}, k_{i,2}, \ldots, k_{i,n_\theta})$,

where scalar kernels $k_{i,j} : [0, T^\star) \times [0, T^\star) \to \mathbb{R}$ are designed for each system matrix element $\Omega_{i,j}$. This kernel design generalizes Rizvi et al. (2018) where the same kernel is used for each element, i.e., $\bar{K}_i = k_i \mathbb{I}$.

**Remark 7.2.** *The matrix-valued kernel function $\bar{K}_i(\cdot,\cdot)$ can also be directly designed as a full matrix to model correlations between the elements in $\Omega_i$ (Álvarez et al., 2012). However, this is beyond the scope of this thesis.*

Then, the predictions on all training points can be expressed as $[\hat{\zeta}_i(t_1)\ \hat{\zeta}_i(t_2)\ \dots\ \hat{\zeta}_i(t_N)]^\top =: \Upsilon_i \alpha_i$, where the $(k,k')$-th element of $\Upsilon_i \in \mathbb{S}_+^N$ is constructed as $(\Upsilon_i)_{k,k'} = \theta(t_k)^\top \bar{K}_i(\tau(t_k), \tau(t_{k'}))\theta(t_{k'})$. Define the collection of state derivative measurements as $Z_i := [\zeta_i(t_1)\ \zeta_i(t_2)\ \dots\ \zeta_i(t_N)]^\top$. The solution to problem (7.15) can then be indirectly given by the closed-form solution of $\alpha_i$:

$$\alpha_i = \bar{\Upsilon}_i^{-1} Z_i, \quad \text{where } \bar{\Upsilon}_i := \Upsilon_i + \lambda_i \mathbb{I} \tag{7.20}$$

through the transformation (7.17). Finally, the system matrix estimates are retrieved as

$$\hat{\Omega}_i(\tau) = \sum_{k=1}^{N} \alpha_{i,k} \theta(t_k)^\top \bar{K}_i(\tau(t_k), \tau). \tag{7.21}$$

**Remark 7.3.** *As discussed in Section 3.1, the identified system matrix function* (7.21) *can also be interpreted as 1) the MAP estimate with the prior knowledge that $\Omega_i(\tau)$ is sampled from a GP with covariance function $\bar{K}_i(\cdot,\cdot)$, or 2) the solution to the regularized function learning problem within the RKHS associated with the kernel $\bar{K}_i(\cdot,\cdot)$, denoted by $\mathcal{H}_{\bar{K}_i}$ (Schölkopf, 2001):*

$$\min_{\Omega_i \in \mathcal{H}_{\bar{K}_i}} \sum_{k=1}^{N} \left(\zeta_i(t_k) - \Omega_i(\tau(t_k))\theta(t_k)\right)^2 + \lambda_i ||\Omega_i||^2_{\mathcal{H}_{\bar{K}_i}}. \tag{7.22}$$

### 7.2.2 Periodic Kernel Design

Since the system matrices are periodic, the periodic kernel design first proposed in MacKay (1998) is used to design $k_{i,j}$. Periodic kernels of period $T^\star$ can be constructed by applying the warping $\chi(\tau) = \left[\sin(\frac{2\pi}{T^\star}\tau)\ \cos(\frac{2\pi}{T^\star}\tau)\right]^\top$ to any standard kernel. For example, consider the squared exponential (SE) kernel described by

$$k_{i,j}^{\mathrm{SE}}(\tau, \tau') := \exp\left(-\frac{||\tau - \tau'||_2^2}{2l_{i,j}^2}\right), \tag{7.23}$$

where $l_{i,j}$ are the hyperparameters, known as the length scale, which control the smoothness of the function estimates. The corresponding periodic kernel is then obtained by substituting $\tau$ and $\tau'$ with $\chi(\tau)$ and $\chi(\tau')$, respectively, and rearranging using trigonometric identities:

$$k_{i,j}^{\mathrm{PSE}}(\tau, \tau') := \exp\left(-\frac{2\sin^2(\frac{\pi}{T^\star}(\tau - \tau'))}{l_{i,j}^2}\right). \tag{7.24}$$

Note that for any $\tau - \tau' = mT^*$, $m \in \mathbb{Z}$, $k_{i,j}^{PSE}(\tau, \tau') = 1$. This means that the function values at $\tau$ and $\tau'$ are perfectly correlated, so the functions learned with such kernels are periodic with period $T^\star$.

The maximum marginal likelihood method introduced in Section 3.2 is again used to identify the hyperparameters in this problem (Rasmussen and Williams, 2006), which are the length scales $l_i := [l_{i,1} \ \dots \ l_{i,n_\theta}]^\top \in \mathbb{R}^{n_\theta}$ associated with each kernel and the regularization parameters $\lambda_i$:

$$\min_{l_i, \lambda_i} \ -\log p(Z_i | \{\theta(t_k), \tau(t_k)\}_{k=1}^N, l_i, \lambda_i), \tag{7.25}$$

where the logarithmic marginal likelihood function is given by

$$p(Z_i | \{\theta(t_k), \tau(t_k)\}_{k=1}^N, l_i, \lambda_i) = \exp\left(-\frac{1}{2} Z_i^\top \bar{\Upsilon}_i^{-1} Z_i - \frac{1}{2} \text{logdet} \bar{\Upsilon}_i + \text{const.}\right). \tag{7.26}$$

### 7.2.3 Extension to Additional Model Parameters

The above identification method can be extended to the case where the system is operated around different operating points, such that the dynamics are also parameter-varying with additional parameter $p$:

$$\dot{x} = f(x, d; p). \tag{7.27}$$

In terms of the transverse dynamics, (7.27) implies an additional dependence on $p$ for the limit cycle $x^\star(\tau, p)$ and the linearized model $\zeta = \Omega(\tau, p)\theta$. The kernel method provides a straightforward way to incorporate such dependence in identification as multivariate functions can be learned by multiplying kernels (Rasmussen and Williams, 2006). In our case, to model the dependence on $p$, the periodic kernel can be multiplied with a standard kernel. For example, for SE kernel, the following multiple kernel can be designed:

$$k^{\text{Multi}}\left(\begin{bmatrix}\tau \\ p\end{bmatrix}, \begin{bmatrix}\tau' \\ p'\end{bmatrix}\right) := k^{\text{PSE}}(\tau, \tau') k^{\text{SE}}(p, p'). \tag{7.28}$$

The proposed identification algorithm is summarized in Algorithm 7.1.

## 7.3 Numerical Results

### 7.3.1 Van der Pol System

The identification algorithm is first tested on the Van der Pol system (7.10) described in Example 7.1. Two sets of data, $\mathcal{D}_1$ and $\mathcal{D}_2$, are generated for identification, which contain trajectories starting from $x_\perp(t_0) = 0.1$ and $x_\perp(t_0) = -0.5$, respectively. For both sets, the forcing term is set to $D = 1$ and $\omega = 10\omega^\star$, where $\omega^\star = \frac{2\pi}{T^\star}$, and zero-mean Gaussian noise with an SNR of 40 dB is injected to the state and state time-derivative measurements. Center transversal surfaces with

**Algorithm 7.1** Kernel-based identification of local limit cycle dynamics with LPPV models

1: **Input:** training data $\{x(t_k), \dot{x}(t_k), d(t_k)\}_{k=1}^{N}$, limit cycle $\Gamma_f^\star$
2: Select transversal surfaces $S(\tau)$ and construct corresponding projection operators $\Pi(\tau)$.
3: Find $\{x_\perp(t_k), \tau(t_k)\}_{k=1}^{N}$ by (7.6) and (7.2).
4: Find $\{\dot{x}_\perp(t_k), \dot{\tau}(t_k)\}_{k=1}^{N}$ by Theorem 1 in Manchester (2011).
5: **for** $i = 1$ **to** $n_x$ **do**
6: **begin**
7: Find $l_i, \lambda_i$ by solving (7.25) with kernel design (7.24).
8: Find $\hat{\Omega}_i(\tau)$ by (7.20) and (7.21).
9: **end**
10: **Output:** transverse system matrix estimate $\hat{\Omega}$

$x_c = \mathbf{0}$ are used. In this example, the computation time is around 4 s on an Intel Core i7-9750H processor at 2.60GHz, which is dominated by the hyperparameter estimation step (7.25).

Figure 7.2 displays the identified system functions from $\mathcal{D}_1$ and $\mathcal{D}_2$, denoted by $\hat{\Omega}(\tau)^{(1)}$ and $\hat{\Omega}(\tau)^{(2)}$, respectively, alongside the analytical transverse linear system functions derived from linearizing the nonlinear system ODE's, indicated by $\Omega(\tau)$. For $\mathcal{D}_1$, where the training data are close to the limit cycle, the identified model matches the analytical one linearized at $x_\perp = 0$ well. Predictions on a test trajectory with $x_\perp(t_0) = -0.5$, $\tau(t_0) = 1.5$, $D = 0.5$, $\omega = 20\omega^\star$ are shown in Figure 7.3 (a) in the phase space, and (b) as time series plots of $x_\perp$ and $(\tau - t)$. It is observed that $\hat{\Omega}(\tau)^{(2)}$ outperforms the other models in terms of prediction error since the trajectory to be predicted is close to the dataset $\mathcal{D}_2$. This indicates that the identification performance improves when the training data is chosen close to the regions where the predictions are to be made and can even be superior to an analytical linearization with a known nonlinear model.

### 7.3.2 Airborne Wind Energy System

Tethered kites are a type of novel power generation systems that exploit the aerodynamic lift generated by the wind. This idea was first proposed in Loyd (1980), and there have been many experimental developments on this topic recently (Ahrens et al., 2013). Under a periodic reference trajectory designed for optimal energy generation, the whole closed-loop system can be considered a periodic system.

A tethered kite system with ground-based power generation during the traction phase is investigated as a physical system example, as illustrated in Figure 7.4. The kite's position is expressed by the elevation angle $\theta$, the azimuth angle $\phi$, and the line length $r$. The unicycle kinematic model from Wood et al. (2015) is considered:

$$\dot{\theta} = \frac{v}{r}\cos(\gamma), \quad \dot{\phi} = \frac{v}{r\cos(\theta)}\sin(\gamma), \quad \dot{\gamma} = u. \tag{7.29}$$

where $x = [\theta \ \phi \ \gamma]^\top$ are the state variables and $u$ is the steering input. The parameters $v$ and $r$ are

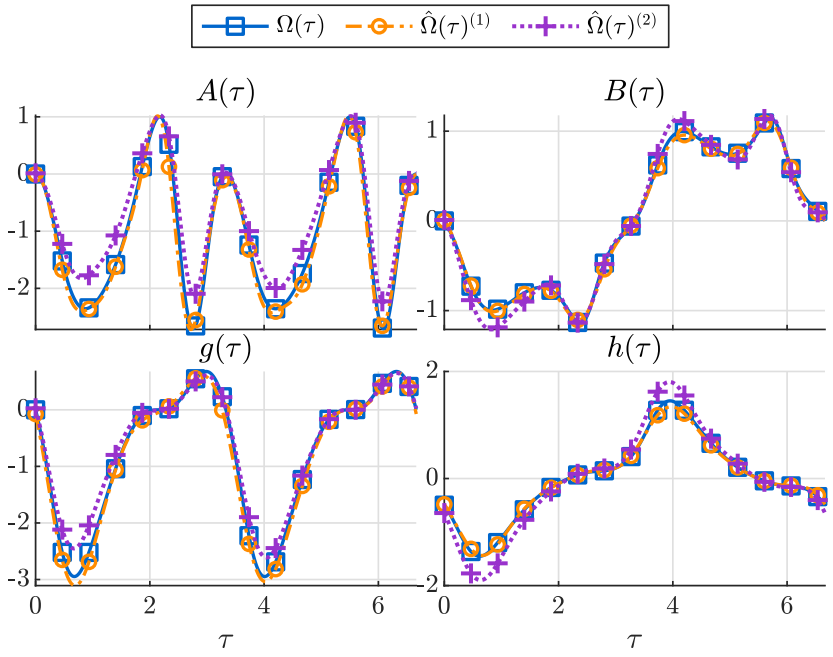



Figure 7.2: Comparison of the identified LPPV models for the Van der Pol system using different training datasets. $\Omega(\tau)$: analytical model, $\hat{\Omega}(\tau)^{(1)}$, $\hat{\Omega}(\tau)^{(2)}$: identified models using $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively.

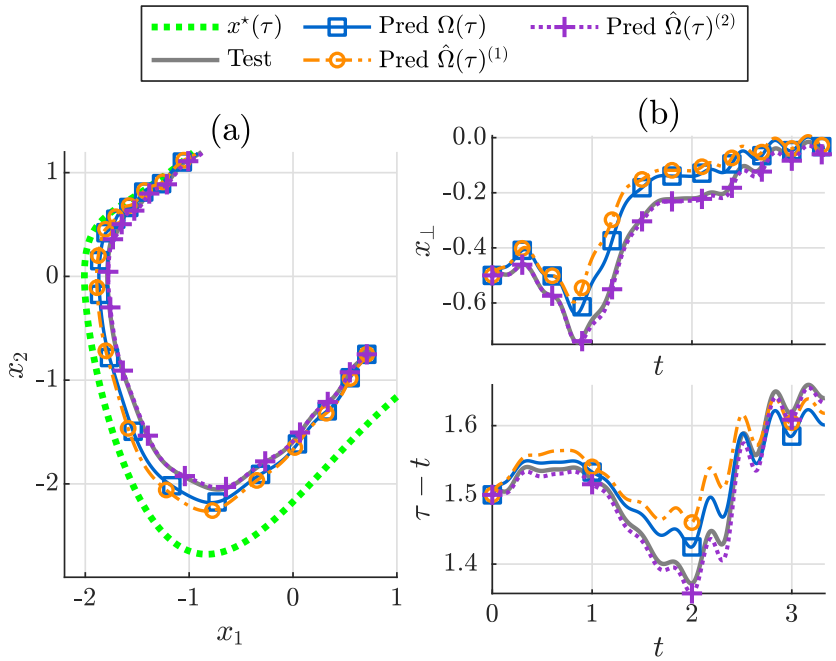



Figure 7.3: Trajectory prediction results of the Van der Pol system, shown (a) in the phase space, and (b) as time series plots of $x_\perp$ and $(\tau - t)$.

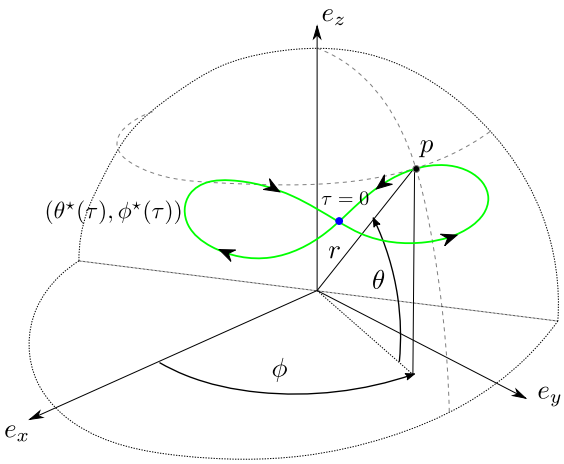



Figure 7.4: Illustration of the tethered kite system and its state variables (Ozan, 2021).

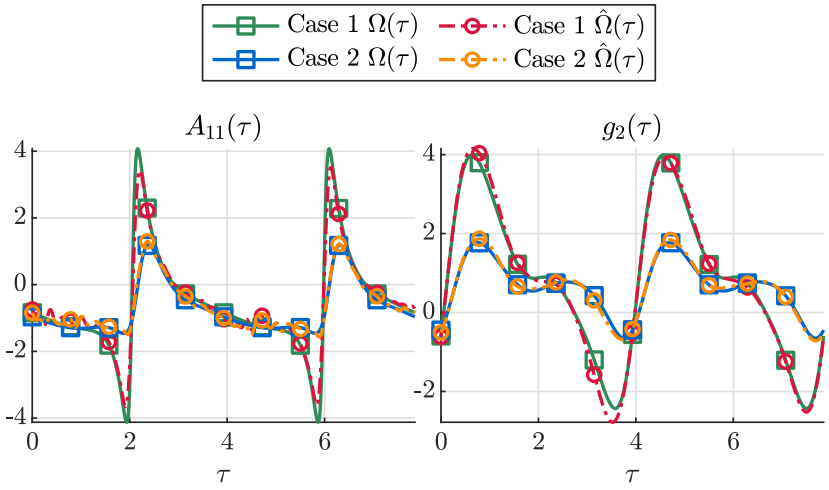



Figure 7.5: Identified LPPV models for the tethered kite system with $\frac{v}{r}$ parametrization. Case 1: $\frac{v}{r} = 0.11$, case 2: $\frac{v}{r} = 0.27$.

assumed to be constant over one cycle. The kite is controlled on a figure-of-eight path for efficient power generation by setting $\gamma^\star(\tau) = a\cos(\omega^\star\tau + b)$, where the frequency $\omega^\star$, the amplitude $a$, and the phase $b$ are determined from the desired midpoint angles and system dynamics (Wood et al., 2015). The control law is designed as transverse state-feedback following Manchester (2011); Ahbe et al. (2018):

$$u(\tau) = u^\star(\tau) + u_\perp(\tau) = u^\star(\tau) - K^\star(\tau)x_\perp(\tau). \tag{7.30}$$

The nominal control input $u^\star(\tau)$ and the controller gains $K^\star(\tau)$ are computed off-line, and a periodically time-varying LQR controller is designed using the linearized periodic system matrix $A(\tau)$ with $Q = \mathbb{I}$, $R = 1$. The associated periodic differential Riccati equation (Bittanti et al., 1991) is solved with the one-shot algorithm (Johansson et al., 2007).

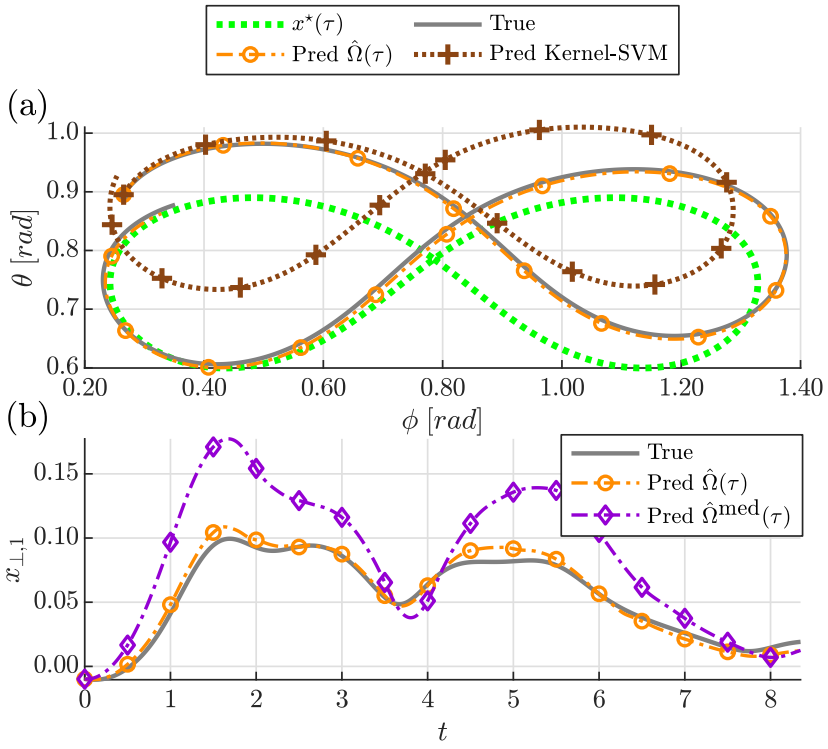



Figure 7.6: Trajectory prediction results of the tethered kite system for $\frac{v}{r} = 0.27$, shown (a) in the phase space of $\theta$ and $\phi$, and (b) as time series plots of $x_{\perp,1}$. Pred $\hat{\Omega}(\tau)$: identified multivariate model, Pred $\hat{\Omega}^{\text{med}}(\tau)$: identified model without $\frac{v}{r}$ parametrization.

The kite system is simulated with a limit cycle of $\omega^\star = 0.8$, $\theta^\star(0) = \frac{\pi}{4}$, $\phi^\star(0) = \frac{\pi}{4}$. During the traction phase, the line length $r$ and the kite velocity $v$ change as the line reels out. As a result, the model parameter $\frac{v}{r}$ varies during operation, and both the limit cycle and the dynamics around it would alter. The variations with respect to $\frac{v}{r}$ can be captured by modifying the periodic SE kernel to the multivariate case as described in Section 7.2.3. Algorithm 7.1 with the extended kernel design (7.28) is applied on trajectory data from different operating points ($\frac{v}{r} \in \{0.3, 0.2154, 0.1625, 0.1263, 0.1\}$), where the training dataset consists of 16 trajectories from random initial conditions with $||x_\perp(t_0)||_2 = 0.02$. Zero-mean Gaussian noise is added to the state and state time-derivative measurements with an SNR of 60 dB. No exogenous input is applied, i.e., $d = 0$. The computation time in this example is around 1080 s.

Figure 7.5 displays the identified models for two parameter values not used in training: $\frac{v}{r} = 0.11$ (case 1) and $\frac{v}{r} = 0.27$ (case 2), with $A_{11}(\tau)$ and $g_2(\tau)$ as examples. The estimates $\hat{\Omega}(\tau)$ are very close to the analytically linearized functions $\Omega(\tau)$. A prediction trajectory is generated for case 2 from a random initial condition with $||x_\perp(t_0)||_2 = 0.1$. Figure 7.6(a) shows the predictions in the phase space of $\theta$ and $\phi$ using the identified model. For benchmarking, the prediction using a black-box kernel support vector machine (SVM) model, which directly learns the nonlinear function (7.27), is also compared. The proposed method accurately predicts the true nonlinear trajectory and performs significantly better than the black-box kernel-SVM method without the

knowledge of the limit cycle. In Figure 7.6(b), the identified model $\hat{\Omega}(\tau)$ is further compared with a model $\hat{\Omega}^{\text{med}}(\tau)$ identified only from the data at $\frac{v}{r} = 0.1625$. The multivariate model obtains better predictions than the model without $\frac{v}{r}$ parametrization.

## 7.4 Summary

A new kernel-based method to identify the local limit cycle dynamics is presented in this chapter. By decomposing the dynamics onto transverse coordinates, local nonlinear dynamics around the limit cycle can be captured by linear approximation of the transverse dynamics with a linear periodically parameter-varying model. The periodic model parameters are identified using the kernel method with periodic kernel design. This framework can be naturally extended to include model variations due to changing operating conditions by leveraging the flexibility of kernel design.

# 8 Conclusions and Outlook

This thesis investigates how the classical paradigm of system identification and model-based control should evolve in response to the rapid advances in machine learning and data science, as well as the challenges imposed by complex systems with limited domain knowledge. One of the critical steps in this evolution is to develop new tools in automatic control that employ nonparametric and high-dimensional approaches instead of compact parametric models.

In Part I of the thesis, plant models are still identified but with general high-dimensional model structures along with regularization techniques. In Chapter 2, an infinite-dimensional sparse learning problem is formulated by characterizing the sparse pole locations of the system with atomic norm regularization. A computationally tractable algorithm is developed to solve the infinite-dimensional problems, followed by iterative weighting and stability selection to reduce the bias and the false positives of the estimate, respectively. This high-dimensional identification framework provides a promising method for identifying linear models with accurate pole location estimation.

In Chapter 3, direct model complexity control is enforced in kernel-based identification by adopting a novel multiple kernel design with optimal first-order kernels and a sparse hyperprior. Reliable error bounds are also derived for kernel-based identification under the practical scenario that the hyperparameters are unknown. Both contributions enlarge the applicability of models identified using the kernel-based method by providing a low-dimensional model realization and a trustworthy uncertainty model, respectively.

In Part II of the thesis, conventional models are replaced by input-output mapping that directly predicts output trajectories from the signal matrix of collected data. In Chapter 4, two methods to obtain well-defined input-output mapping in the presence of unbounded uncertainties are proposed, by solving a low-rank Hankel matrix denoising problem and a maximum likelihood estimation problem, respectively. Superior noise reduction performance is achieved in matrix denoising by exploiting a data-driven singular value shrinkage law with Hankel approximation. The signal matrix model derived from the maximum likelihood estimation framework provides accurate impulse response estimation with less restrictive assumptions than the conventional

least-squares method. The nominal prediction from the signal matrix model is also augmented with confidence region characterization, providing an uncertainty model. These contributions provide a practical framework to develop and analyze data-driven predictors with stochastic data.

Chapter 5 investigates the application of data-driven predictors to receding horizon predictive control. By adopting the linearized signal matrix model predictor with certainty equivalence, indirect data-driven predictive control demonstrates improved performance compared to subspace predictive control and regularized data-enabled predictive control. The algorithm is extended by including the uncertainty model of the predictor to provide more accurate prediction and guaranteed constraint satisfaction with initial condition estimation and chance constraint tightening, respectively. As illustrated in a space heating control case study, the proposed stochastic indirect data-driven predictive control algorithm shows great potential for providing excellent control performance while satisfying operating constraints in practice.

In Part III of the thesis, the methodologies are extended to periodic systems. Chapter 6 focuses on the identification of linear time-periodic systems. In the time domain, the atomic norm regularization approach is extended to periodic systems by considering grouped coefficients of the reformulated switching models. In the frequency domain, a subspace identification method is developed by investigating the frequency responses of the lifted periodic system model. Both methods demonstrate that periodic systems can be successfully identified by integrating structural constraints in applying identification methods to their linear time-invariant reformulations.

Chapter 7 utilizes linear periodic models to model local limit cycle dynamics of nonlinear systems. By linearizing the nonlinear dynamics along the limit cycle, the local dynamics can be learned as a nonlinear periodic function by kernel learning with a periodic kernel design. This methodology can be employed to characterize the closed-loop performance of periodically operating control systems, as demonstrated by an airborne wind energy example.

The following future research directions are presented as an outlook.

**Bayesian perspective of behavioral system theory**. The Willems' fundamental lemma is based on binary certification of possible system behaviors. However, as mentioned in Chapter 4, no trajectory can be falsified when only stochastic data with unbounded noise are available. Therefore, a Bayesian description may be preferred which describes possible system behaviors as the posterior probability of observing a certain trajectory given the collected data. This perspective leads to a stochastic version of behavioral system theory, which can support a unified framework of direct data-driven denoising, prediction, and control.

**Active Exploration in data-driven predictive control**. Section 5.1.4 demonstrates that data-driven predictors can be adaptive, incorporating online data to enhance control performance. This enables dual-control design in data-driven predictive control by actively exploring regions that result in predictions with high uncertainties. A possible framework in this direction is Bayesian optimization, employing methods like the upper confidence bound policy (Garivier and Moulines, 2011).

**Nonlinear data-driven predictive control via Koopman operator**. Although the Willems' fundamental lemma heavily relies on the linearity assumption, it can be extended to nonlinear systems by leveraging their linear representations. This direction has been explored for Hammerstein-Wiener models (Berberich and Allgöwer, 2020). The Koopman theory can provide linear representations of general nonlinear systems, and data-driven predictive control can be conducted on the Koopman observables. The main difficulties in this direction are 1) how to satisfy the data informativity condition for potentially infinite-dimensional linear representations and 2) how to quantify the prediction error associated with the Koopman operator approximation.

# Bibliography


Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Ahbe, E., Iannelli, A., and Smith, R. S. (2022). A novel moving orthonormal coordinate-based approach for region of attraction analysis of limit cycles. *Journal of Computational Dynamics*.

Ahbe, E., Wood, T. A., and Smith, R. S. (2018). Stability verification for periodic trajectories of autonomous kite power systems. In *European Control Conference (ECC)*, pages 46–51.

Ahrens, U., Diehl, M., and Schmehl, R. (2013). *Airborne Wind Energy*. Springer, Heidelberg, Germany.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Allen, M. S. and Sracic, M. W. (2009). System identification of dynamic systems with cubic nonlinearities using linear time-periodic approximations. In *International Conference on Multibody Systems, Nonlinear Dynamics, and Control*, volume 4, pages 731–741. ASME.

Allen, M. S., Sracic, M. W., Chauhan, S., and Hansen, M. H. (2011). Output-only modal analysis of linear time-periodic systems with application to wind turbine simulation data. *Mechanical Systems and Signal Processing*, 25(4):1174–1191.

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

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 4(3):195–266.

Aravkin, A., Burke, J. V., Chiuso, A., and Pillonetto, G. (2014). Convex vs non-convex estimators for regression and sparse estimation: the mean squared error properties of ARD and GLasso. *The Journal of Machine Learning Research*, 15(1):217–252.

Bachnas, A., Tòth, R., Ludlage, J., and Mesbah, A. (2014). A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study. *Journal of Process Control*, 24:272–285.

Baggio, G., Carè, A., Scampicchio, A., and Pillonetto, G. (2022). Bayesian frequentist bounds for machine learning and system identification. *Automatica*, 146:110599.

Bauer, D. (2001). Order estimation for subspace methods. *Automatica*, 37(10):1561–1573.

Beckers, T., Umlauft, J., and Hirche, S. (2018). Mean square prediction error of misspecified Gaussian process models. In *IEEE Conference on Decision and Control (CDC)*, pages 1162–1167.

Benaych-Georges, F. and Nadakuditi, R. R. (2012). The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135.

Berberich, J. and Allgöwer, F. (2020). A trajectory-based framework for data-driven system analysis and control. In *European Control Conference (ECC)*, pages 1365–1370.

Berberich, J., Kohler, J., Muller, M. A., and Allgower, F. (2020). Robust constraint satisfaction in data-driven MPC. In *IEEE Conference on Decision and Control (CDC)*.

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

Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. (2017). Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, volume 30.

Berkooz, G., Holmes, P., and Lumley, J. L. (1993). The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575.

Bittanti, S. (1986). Deterministic and stochastic linear periodic systems. In *Time Series and Linear Systems*, pages 141–182. Springer-Verlag.

Bittanti, S. and Colaneri, P. (2000). Invariant representations of discrete-time periodic systems. *Automatica*, 36(12):1777–1793.

Bittanti, S. and Colaneri, P. (2009). *Periodic systems: filtering and control*. Springer, London, UK.

Bittanti, S., Colaneri, P., and De Nicolao, G. (1991). *The Periodic Riccati Equation*, pages 127–162. Springer, Heidelberg, Germany.

Boggs, P. T. and Tolle, J. W. (1995). Sequential quadratic programming. *Acta Numerica*, 4(1):1–51.

Bojarski, A., Khayatian, F., and Cai, H. (2023). nestli: Neighborhood Energy System Testing towards Large-scale Integration. https://doi.org/10.5281/zenodo.7635812. [Online; accessed: 25.04.2023].

Brand, M. (2002). Incremental singular value decomposition of uncertain data with missing values. In *Computer Vision — ECCV*, pages 707–720. Springer.

Breschi, V., Chiuso, A., and Formentin, S. (2023). Data-driven predictive control in a stochastic setting: a unified framework. *Automatica*, 152:110961.

Budman, H. and Silveston, P. L. (2013). Control of periodically operated reactors. In *Periodic Operation of Chemical Reactors*, pages 543–567. Elsevier.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer, Heidelberg, Germany.

Cadzow, J. (1988). Signal enhancement-a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(1):49–62.

Campi, M., Lecchini, A., and Savaresi, S. (2002). Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8):1337–1346.

Capone, A., Lederer, A., and Hirche, S. (2022). Gaussian process uniform error bounds with unknown hyperparameters for safety-critical applications. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2609–2624.

Carapia, G. Q., Markovsky, I., Pintelon, R., Csurcsia, P. Z., and Verbeke, D. (2020). Experimental validation of a data-driven step input estimation method for dynamic measurements. *IEEE Transactions on Instrumentation and Measurement*, 69(7):4843–4851.

Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214.

Chen, T. (2018). On kernel design for regularized LTI system identification. *Automatica*, 90:109–122.

Chen, T., Andersen, M. S., Ljung, L., Chiuso, A., and Pillonetto, G. (2014). System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques. *IEEE Transactions on Automatic Control*, 59(11):2933–2945.

Chen, T., Andersen, M. S., Mu, B., Yin, F., Ljung, L., and Qin, S. J. (2018). Regularized LTI system identification with multiple regularization matrix. *IFAC-PapersOnLine*, 51(15):180–185.

Chen, T., Ohlsson, H., and Ljung, L. (2012). On the estimation of transfer functions, regularizations and Gaussian processes—revisited. *Automatica*, 48(8):1525–1535.

Chiuso, A. and Pillonetto, G. (2012). A Bayesian approach to sparse dynamic network identification. *Automatica*, 48(8):1553–1565.

Chiuso, A. and Pillonetto, G. (2019). System identification: a machine learning perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):281–304.

Coulson, J., Lygeros, J., and Dorfler, F. (2022). Distributionally robust chance constrained data-enabled predictive control. *IEEE Transactions on Automatic Control*, 67(7):3289–3304.

Coulson, J., Lygeros, J., and Dörfler, F. (2019). Data-enabled predictive control: In the shallows of the DeePC. In *European Control Conference (ECC)*, pages 307–312.

Cox, P. B. (2018). *Towards efficient identification of linear parameter-varying state-space models*. PhD thesis, Eindhoven University of Technology.

Damen, A., Van den Hof, P., and Hajdasinski, A. (1982). Approximate realization based upon an alternative to the Hankel matrix: the Page matrix. *Systems & Control Letters*, 2(4):202–208.

De Persis, C. and Tesi, P. (2020). Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924.

Dobrowiecki, T. P., Schoukens, J., and Guillaume, P. (2006). Optimized excitation signals for MIMO frequency response function measurements. *IEEE Transactions on Instrumentation and Measurement*, 55(6):2072–2079.

Dörfler, F., Coulson, J., and Markovsky, I. (2023). Bridging direct & indirect data-driven control formulations via regularizations and relaxations. *IEEE Transactions on Automatic Control*, 68(2):883–897.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Elokda, E., Coulson, J., Beuchat, P. N., Lygeros, J., and Dörfler, F. (2021). Data-enabled predictive control for quadcopters. *International Journal of Robust and Nonlinear Control*, 31(18):8916–8936.

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

Fazel, M., Hindi, H., and Boyd, S. (2001). A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference (ACC)*, volume 6, pages 4734–4739. IEEE.

Fazel, M., Hindi, H., and Boyd, S. P. (2003). Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *American Control Conference (ACC)*, volume 3, pages 2156–2162. IEEE.

Felici, F., van Wingerden, J.-W., and Verhaegen, M. (2007). Subspace identification of MIMO LPV systems using a periodic scheduling sequence. *Automatica*, 43(10):1684–1697.

Fiedler, F. and Lucia, S. (2021). On the relationship between data-enabled predictive control and subspace predictive control. In *European Control Conference (ECC)*, pages 222–229.

Fukushima, H., Kim, T.-H., and Sugie, T. (2007). Adaptive model predictive control for a class of constrained linear systems based on the comparison model. *Automatica*, 43(2):301–308.

Furieri, L., Guo, B., Martin, A., and Ferrari-Trecate, G. (2021). A behavioral input-output parametrization of control policies with suboptimality guarantees. In *IEEE Conference on Decision and Control (CDC)*. IEEE.

Furieri, L., Guo, B., Martin, A., and Ferrari-Trecate, G. (2023). Near-optimal design of safe output-feedback controllers from noisy data. *IEEE Transactions on Automatic Control*, 68(5):2699–2714.

Garivier, A. and Moulines, E. (2011). On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pages 174–188.

Gasso, G., Rakotomamonjy, A., and Canu, S. (2009). Recovering sparse signals with a certain family of nonconvex penalties and DC programming. *IEEE Transactions on Signal Processing*, 57(12):4686–4698.

Gavish, M. and Donoho, D. L. (2014). The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053.

Gavish, M. and Donoho, D. L. (2017). Optimal shrinkage of singular values. *IEEE Transactions on Information Theory*, 63(4):2137–2152.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58.

Ghodsi, M., Alharbi, N., and Hassani, H. (2015). The empirical distribution of the singular values of a random hankel matrix. *Fluctuation and Noise Letters*, 14(3):1550027.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU press.

Goos, J. and Pintelon, R. (2014). Continuous time frequency domain LPV state space identification via periodic time-varying input-output modeling. In *IEEE Conference on Decision and Control (CDC)*.

Goos, J. and Pintelon, R. (2016). Continuous-time identification of periodically parameter-varying state space models. *Automatica*, 71:254–263.

Hale, J. K. (1980). *Ordinary Differential Equations*. R.E. Krierger Pub. Co., New York.

Hallouzi, R. and Verhaegen, M. (2008). Fault-tolerant subspace predictive control applied to a Boeing 747 model. *Journal of Guidance, Control, and Dynamics*, 31(4):873–883.

Hench, J. J. (1995). A technique for the identification of linear periodic state-space models. *International Journal of Control*, 62(2):289–301.

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

Hjalmarsson, H., Gevers, M., Gunnarsson, S., and Lequin, O. (1998). Iterative feedback tuning: theory and applications. *IEEE Control Systems Magazine*, 18(4):26–41.

Hong, S., Mu, B., Yin, F., Andersen, M. S., and Chen, T. (2018). Multiple kernel based regularized system identification with SURE hyper-parameter estimator. *IFAC-PapersOnLine*, 51(15):13–18.

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

Huang, L., Coulson, J., Lygeros, J., and Dörfler, F. (2021). Decentralized data-enabled predictive control for power system oscillation damping. *IEEE Transactions on Control Systems Technology*.

Huang, L., Coulson, J., Lygeros, J., and Dörfler, F. (2019). Data-enabled predictive control for grid-connected power converters. In *IEEE Conference on Decision and Control (CDC)*, pages 8130–8135.

Huang, L., Zhen, J., Lygeros, J., and Dörfler, F. (2023). Robust data-enabled predictive control: Tractable formulations and performance guarantees. *IEEE Transactions on Automatic Control*, 68(5):3163–3170.

Iannelli, A., Yin, M., and Smith, R. S. (2021a). Design of input for data-driven simulation with hankel and page matrices. In *IEEE Conference on Decision and Control (CDC)*, pages 139–145.

Iannelli, A., Yin, M., and Smith, R. S. (2021b). Experiment design for impulse response identification with signal matrix models. *IFAC-PapersOnLine*, 54(7):625–630.

Johansson, S., Kågström, B., Shiriaev, A., and Varga, A. (2007). Comparing one-shot and multi-shot methods for solving periodic Riccati differential equations. *IFAC Proceedings Volumes*, 3(1):163–168.

Josse, J. and Sardy, S. (2015). Adaptive shrinkage of singular values. *Statistics and Computing*, 26(3):715–724.

Kadali, R., Huang, B., and Rossiter, A. (2003). A data driven subspace approach to predictive controller design. *Control Engineering Practice*, 11(3):261–278.

Kergus, P. and Gosea, I. V. (2022). Data-driven approximation and reduction from noisy data in matrix pencils frameworks. *IFAC-PapersOnLine*, 55(30):371–376.

Kerz, S., Teutsch, J., Brüdigam, T., Leibold, M., and Wollherr, D. (2023). Data-driven tube-based stochastic predictive control. *IEEE Open Journal of Control Systems*, 2:185–199.

Khayatian, F., Cai, H., Bojarski, A., Heer, P., and Bollinger, A. (2022). Benchmarking HVAC controller performance with a digital twin. In *Applied Energy Symposium: Clean Energy towards Carbon Neutrality (CEN)*.

Khosravi, M. (2021). *Side-Information in Linear and Nonlinear System Identification*. Doctoral thesis, ETH Zurich, Zurich.

Khosravi, M., Eichler, A., and Smith, R. S. (2017). Automated classification and identification procedure for prediction of energy consumption in multi-mode buildings. *Energy Procedia*, 122:1021–1026.

Klöppelt, C., Berberich, J., Allgöwer, F., and Müller, M. A. (2022). A novel constraint-tightening approach for robust data-driven predictive control. *International Journal of Robust and Nonlinear Control*.

Kouvaritakis, B. and Cannon, M. (2016). *Model Predictive Control: Classical, Robust and Stochastic*. Springer, Cham, Switzerland.

Lagoudakis, M. G. and Parr, R. (2003). Least-squares policy iteration. *Journal of machine learning research*, 4(Dec):1107–1149.

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

Laurain, V., Tóth, R., Zheng, W.-X., and Gilson, M. (2012). Nonparametric identification of LPV models under general noise conditions: An LS-SVM based approach. *IFAC Proceedings Volumes*, 45(16):1761–1766. 16th IFAC Symposium on System Identification.

Li, Y., Liu, K. R., and Razavilar, J. (1997). A parameter estimation scheme for damped sinusoidal signals based on low-rank hankel approximation. *IEEE Transactions on Signal Processing*, 45(2):481–486.

Lian, Y., Shi, J., Koch, M., and Jones, C. N. (2023). Adaptive robust data-driven building control via bilevel reformulation: An experimental result. *IEEE Transactions on Control Systems Technology*, 31(6):2420–2436.

Liu, K. (1997). Identification of linear time-varying systems. *Journal of Sound and Vibration*, 206(4):487–505.

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

Ljung, L., Chen, T., and Mu, B. (2019). A shift in paradigm for system identification. *International Journal of Control*, 93(2):173–180.

Louarroudi, E., Pintelon, R., and Lataire, J. (2012). Nonparametric tracking of the time-varying dynamics of weakly nonlinear periodically time-varying systems using periodic inputs. *IEEE Transactions on Instrumentation and Measurement*, 61(5):1384–1394.

Loyd, M. L. (1980). Crosswind kite power (for large-scale wind power production). *Journal of Energy*, 4(3):106–111.

MacKay, D. J. (1998). Introduction to Gaussian processes. *NATO ASI series F: computer and systems sciences*, 168:133–166.

Maddalena, E. T., Scharnhorst, P., and Jones, C. N. (2021). Deterministic error bounds for kernel-based learning techniques under bounded noise. *Automatica*, 134:109896.

Manchester, I. R. (2011). Transverse dynamics and regions of stability for nonlinear hybrid limit cycles. *IFAC Proceedings Volumes*, 44(1):6285–6290. 18th IFAC World Congress.

Marchenko, V. A. and Pastur, L. A. (1967). Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483.

Marconato, A., Schoukens, M., and Schoukens, J. (2016). Filter-based regularisation for impulse response modelling. *IET Control Theory & Applications*, 11:194–204.

Markovsky, I. and Dörfler, F. (2021). Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, 52:42–64.

Markovsky, I. and Dörfler, F. (2023). Identifiability in the behavioral setting. *IEEE Transactions on Automatic Control*, 68(3):1667–1677.

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

Markovsky, I. and Usevich, K. (2013). Structured low-rank approximation with missing data. *SIAM Journal on Matrix Analysis and Applications*, 34(2):814–830.

Markovsky, I. and Usevich, K. (2014). Software for weighted structured low-rank approximation. *Journal of Computational and Applied Mathematics*, 256:278–292.

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

Markovsky, I., Willems, J. C., Rapisarda, P., and Moor, B. L. D. (2005b). Data driven simulation with application to system identification. *IFAC Proceedings Volumes*, 38(1):970–975.

McKelvey, T., Akçay, H., and Ljung, L. (1996). Subspace-based identification of infinite-dimensional multivariable systems from frequency-response data. *Automatica*, 32(6):885–902.

Meinshausen, N., Meier, L., and Bühlmann, P. (2009). p-values for high-dimensional regression. *Journal of the American Statistical Association*, 104(488):1671–1681.

Moonen, M., De Moor, B., Vandenberghe, L., and Vandewalle, J. (1989). On- and off-line identification of linear state-space models. *International Journal of Control*, 49(1):219–232.

Mu, B., Chen, T., and Ljung, L. (2018a). Asymptotic properties of generalized cross validation estimators for regularized system identification. *IFAC-PapersOnLine*, 51(15):203–208.

Mu, B., Chen, T., and Ljung, L. (2018b). Asymptotic properties of hyperparameter estimators by using cross-validations for regularized system identification. In *IEEE Conference on Decision and Control (CDC)*, pages 644–649.

Möllerstedt, E. and Bernhardsson, B. (2000). Out of control because of harmonics-an analysis of the harmonic response of an inverter locomotive. *IEEE Control Systems*, 20(4):70–81.

Nadakuditi, R. R. (2014). OptShrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage. *IEEE Transactions on Information Theory*, 60(5):3002–3018.

Ohlsson, H. and Ljung, L. (2013). Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49(4):1045–1050.

Oldewurtel, F., Jones, C. N., and Morari, M. (2008). A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback. In *IEEE Conference on Decision and Control (CDC)*.

Oldewurtel, F., Parisio, A., Jones, C. N., Gyalistras, D., Gwerder, M., Stauch, V., Lehmann, B., and Morari, M. (2012). Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45:15–27.

Overschee, P. V. and Moor, B. D. (1996). *Subspace Identification for Linear Systems*. Springer US.

Ozan, D. E. (2021). Identification of limit cycle dynamics with linear periodic models for airborne wind energy applications. Master thesis, ETH Zurich, Zurich, Switzerland.

Pan, G., Ou, R., and Faulwasser, T. (2023). Data-driven stochastic output-feedback predictive control: Recursive feasibility through interpolated initial conditions. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, pages 980–992.

Pan, W., Yuan, Y., Ljung, L., Goncalves, J., and Stan, G.-B. (2018). Identification of nonlinear state-space systems from heterogeneous datasets. *IEEE Transactions on Control of Network Systems*, 5(2):737–747.

Park, H., Zhang, L., and Rosen, J. B. (1999). Low rank approximation of a hankel matrix by structured total least norm. *BIT Numerical Mathematics*, 39(4):757–779.

Pillonetto, G., Chen, T., Chiuso, A., De Nicolao, G., and Ljung, L. (2022). *Regularized system identification: learning dynamic models from data*. Springer.

Pillonetto, G., Chen, T., Chiuso, A., Nicolao, G. D., and Ljung, L. (2016). Regularized linear system identification using atomic, nuclear and kernel-based norms: the role of the stability constraint. *Automatica*, 69:137–149.

Pillonetto, G. and De Nicolao, G. (2010). A new kernel-based approach for linear system identification. *Automatica*, 46(1):81–93.

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

Pillonetto, G. and Ljung, L. (2023). Full bayesian identification of linear dynamic systems using stable kernels. *Proceedings of the National Academy of Sciences*, 120(18).

Pillonetto, G. and Scampicchio, A. (2022). Sample complexity and minimax properties of exponentially stable regularized estimators. *IEEE Transactions on Automatic Control*, 67(5):2330–2342.

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. Wiley.

Rakotomamonjy, A., Flamary, R., and Yger, F. (2012). Learning with infinitely many features. *Machine Learning*, 91(1):43–66.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Massachussets.

Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):253–279.

Rizvi, S. Z., Velni, J. M., Abbasi, F., Tòth, R., and Meskin, N. (2018). State-space LPV model identification using kernelized machine learning. *Automatica*, 88:38–47.

Rosset, S., Swirszcz, G., Srebro, N., and Zhu, J. (2007). $l_1$ regularization in infinite dimensional feature spaces. In *Learning Theory*, pages 544–558, Heidelberg, Germany. Springer.

Safonov, M. G. and Tung-Ching Tsao (1997). The unfalsified control concept and learning. *IEEE Transactions on Automatic Control*, 42(6):843–847.

Saitoh, S. and Sawano, Y. (2016). *Theory of reproducing kernels and applications*. Springer, Singapore.

Schölkopf, B. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press, Cambridge, Massachusetts.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *Computational Learning Theory*, pages 416–426.

Schoukens, J. and Ljung, L. (2019). Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, 39(6):28–99.

Schoukens, J., Pintelon, R., and Guillaume, P. (1994). On the advantages of periodic excitation in system identification. *IFAC Proceedings Volumes*, 27(8):1115–1120.

Sedghizadeh, S. and Beheshti, S. (2018). Data-driven subspace predictive control: Stability and horizon tuning. *Journal of the Franklin Institute*, 355(15):7509–7547.

Sefidmazgi, M. G., Kordmahalleh, M. M., Homaifar, A., Karimoddini, A., and Tunstel, E. (2016). A bounded switching approach for identification of switched MIMO systems. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*.

Shah, P., Bhaskar, B. N., Tang, G., and Recht, B. (2012). Linear system identification via atomic norm regularization. In *IEEE Conference on Decision and Control (CDC)*, pages 6265–6270. IEEE.

Shah, R. D. and Samworth, R. J. (2013). Variable selection with error control: another look at stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(1):55–80.

Shen, X., Diamond, S., Gu, Y., and Boyd, S. (2016). Disciplined convex-concave programming. In *IEEE Conference on Decision and Control (CDC)*, pages 1009–1014.

Shin, S. J., Cesnik, C. E. S., and Hall, S. R. (2005). System identification technique for active helicopter rotors. *Journal of Intelligent Material Systems and Structures*, 16(11-12):1025–1038.

Smith, R. S. (2014). Frequency domain subspace identification using nuclear norm minimization and Hankel matrix realizations. *IEEE Transactions on Automatic Control*, 59(11):2886–2896.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. (2012). Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265.

Stoica, P., Eykhoff, P., Janssen, P., and Söderström, T. (1986). Model-structure selection by cross-validation. *International Journal of Control*, 43(6):1841–1878.

Strogatz, S. H. (1994). *Nonlinear dynamics and chaos with applications to physics, biology, chemistry and engineering*. Addison-Wesley, Reading, Massachusetts.

Sutton, R. (2019). The bitter lesson. Blog: http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tóth, R. (2010). *Modeling and identification of linear parameter-varying systems*, volume 403. Springer, Heidelberg, Germany.

Tuo, R. and Wang, W. (2022). Kriging prediction with isotropic Matérn correlations: Robustness and experimental designs. *The Journal of Machine Learning Research*, 21(1):7604–7641.

Tóth, R., Laurain, V., Zheng, W. X., and Poolla, K. (2011). Model structure learning: A support vector machine approach for LPV linear-regression models. In *IEEE Conference on Decision and Control and European Control Conference*, pages 3192–3197.

Umlauft, J., Beckers, T., Kimmel, M., and Hirche, S. (2017). Feedback linearization using Gaussian processes. In *IEEE Conference on Decision and Control (CDC)*.

Uyanik, I., Saranli, U., Ankarali, M. M., Cowan, N. J., and Morgul, O. (2019). Frequency-domain subspace identification of linear time-periodic (LTP) systems. *IEEE Transactions on Automatic Control*, 64(6):2529–2536.

van Waarde, H. J., Camlibel, M. K., and Mesbahi, M. (2022). From noisy data to feedback controllers: Nonconservative design via a matrix s-lemma. *IEEE Transactions on Automatic Control*, 67(1):162–175.

van Waarde, H. J., De Persis, C., Camlibel, M. K., and Tesi, P. (2020). Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Systems Letters*, 4(3):602–607.

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

Verhaegen, M. and Yu, X. (1995). A class of subspace model identification algorithms to identify periodically and arbitrarily time-varying systems. *Automatica*, 31(2):201–216.

Wang, C., Zhu, Z., Gu, H., Wu, X., and Liu, S. (2019). Hankel low-rank approximation for seismic noise attenuation. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1):561–573.

Wang, H. and Leng, C. (2008). A note on adaptive group lasso. *Computational Statistics & Data Analysis*, 52(12):5277–5286.

Wereley, N. M. (1990). *Analysis and control of linear periodically time varying systems*. PhD thesis, Massachusetts Institute of Technology.

Willems, J. C. and Polderman, J. W. (1997). *Introduction to Mathematical Systems Theory: A Behavioral Approach*, volume 26. Springer, New York, NY, USA.

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

Wood, T. A., Hesse, H., Polzin, M., Ahbe, E., and Smith, R. S. (2018). Modeling, identification, estimation and adaptation for the control of power-generating kites. *IFAC Symposium on System Identification, IFAC-PapersOnLine*, 51(15):981–989.

Wood, T. A., Hesse, H., Zgraggen, A. U., and Smith, R. S. (2015). Model-based flight path planning and tracking for tethered wings. In *IEEE Conference on Decision and Control (CDC)*, pages 6712–6717.

Yen, I. E. H., Lin, T. W., Lin, S. D., Ravikumar, P. K., and Dhillon, I. S. (2014). Sparse random feature algorithm as coordinate descent in Hilbert space. In *Advances in Neural Information Processing Systems*, volume 27.

Yin, W. and Mehr, A. S. (2009). Identification of linear periodically time-varying systems using periodic sequences. In *IEEE International Conference on Control Applications*. IEEE.

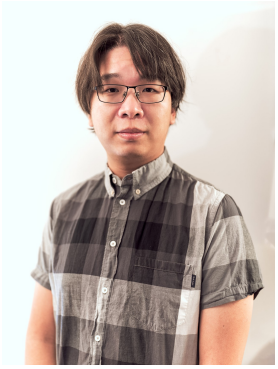
Yu, Y., Talebi, S., van Waarde, H. J., Topcu, U., Mesbahi, M., and Acikmese, B. (2021). On controllability and persistency of excitation in data-driven control: Extensions of willems' fundamental lemma. In *IEEE Conference on Decision and Control (CDC)*. IEEE.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Yuille, A. L. and Rangarajan, A. (2002). The concave-convex procedure (CCCP). In *Advances in Neural Information Processing Systems*, pages 1033–1040.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Åström, K. (1980). Maximum likelihood and prediction error methods. *Automatica*, 16(5):551–574.



# Mingzhou Yin

7-1-201, Zhongshanbei Road 200-2
210009 Nanjing, Jiangsu
China

Phone: +86 136 4515 7032
Email: mingzhouyin@gmail.com
Web: https://mingzhouyin.github.io/

---

### Brief biography

Mingzhou Yin has been a doctoral student supervised by Professor Roy S. Smith in the Automatic Control Laboratory at ETH Zurich since February 2019. His doctoral thesis is on regularized and nonparametric approaches in system identification and data-driven control. He has been supported by the project "Modeling, Identification and Control of Periodic Systems in Energy Applications" from the Swiss National Science Foundation. He received his MSc degree *cum laude* in control & simulation at the Faculty of Aerospace Engineering, Delft University of Technology, the Netherlands in 2018. His master's thesis research is on envelope-protected non-linear control of over-actuated aircraft in collaboration with Lockheed Martin. He received the joint bachelor's degree in Mechanical Engineering at Shanghai Jiao Tong University, China, and the University of Hong Kong, China with first-class honours in 2016. He was the recipient of the IEEE Control Systems Society Swiss Chapter Young Author Best Journal Paper Award and the Systems Identification and Adaptive Control Technical Committee Outstanding Student Paper Prize in 2023.

---

### Research interests

His research interests include data-driven modeling, simulation & control, sparse learning theory, system identification with subspace and regularized methods, model predictive control, and periodic system theory.

---

### Education

**2019.02 - 2024.02** Doctor of Sciences, Automatic Control Laboratory, ETH Zurich, Switzerland

Advisor: Roy S. Smith

Thesis: Regularized and nonparametric approaches in system identification and data-driven control

**2016.09 – 2018.07** Master of Science in Aerospace Engineering [*Cum Laude*], Delft University of Technology, the Netherlands

Advisors: Coen de Visser, Qiping Chu

Thesis: Envelope estimation and protection of innovative control effectors (ICE) aircraft

**2012.09 - 2016.06** Bachelor of Engineering in Mechanical Engineering [First Class Honours], Shanghai Jiao Tong University, China & the University of Hong Kong, China

---

**Awards**

**2023** IEEE Control Systems Society Swiss Chapter Young Author Best Journal Paper Award

**2023** IEEE Control Systems Society Systems Identification and Adaptive Control Technical Committee Outstanding Student Paper Prize

---

**Publications**

Journals

- **Yin M.**, Cai H., Gattiglio A., Khayatian F., Smith R.S, Heer P. (2024). Data-driven Predictive Control for Demand Side Management: Theoretical and Experimental Results. *Applied Energy*, 353(A), 122101.
- **Yin M.**, Smith R.S. (2023). Error Bounds for Kernel-Based Linear System Identification with Unknown Hyperparameters. *IEEE Control Systems Letters*, 7, 2491-2496.
- **Yin M.**, Iannelli A., Smith R.S. (2021). Maximum Likelihood Estimation in Data-Driven Modeling and Control. *IEEE Transactions on Automatic Control*, 68(1), 317-328.
- **Yin M.**, Iannelli A., Smith R.S. (2021). Subspace Identification of Linear Time-Periodic Systems with Periodic Inputs. *IEEE Control Systems Letters*, 5(1), 145-150.
- **Yin M.**, Chu Q. P., Zhang Y., Niestroy M. A., de Visser C. C. (2019). Probabilistic Flight Envelope Estimation with Application to Unstable Overactuated Aircraft. *Journal of Guidance, Control, and Dynamics*, 42(12), 2650-2663.

Conference Papers

- Srivastava A., **Yin M.**, Iannelli A., Smith R.S. (2023). A Dual System-Level Parameterization for Identification from Closed-Loop Data. *IEEE Conference on Decision and Control*.
- **Yin M.**, Akan M.T., Iannelli A., Smith R.S. (2022). Infinite-Dimensional Sparse Learning in Linear System Identification. *IEEE Conference on Decision and Control*.
- Ozan D.E., **Yin M.**, Iannelli A., Smith R.S. (2022). Kernel-Based Identification of Local Limit Cycle Dynamics with Linear Periodically Parameter-Varying Models. *IEEE Conference on Decision and Control*.
- **Yin M.**, Iannelli A., Smith R.S. (2022). Data-Driven Prediction with Stochastic Data: Confidence Regions and Minimum Mean-Squared Error Estimates. *European Control Conference*.
- Iannelli A., **Yin M.**, Smith R.S. (2021). Design of Input for Data-Driven Simulation with Hankel and Page Matrices. *IEEE Conference on Decision and Control*.
- Ozan, D.E., Iannelli A., **Yin M.**, Smith R.S. (2021). Regularized Classification and Simulation of Bifurcation Regimes in Nonlinear Systems. *The 3rd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems*.
- Iannelli A., **Yin M.**, Smith R.S. (2021). Experiment Design for Impulse Response Identification with Signal Matrix Models. *The 19th IFAC Symposium on System Identification*.
- **Yin M.**, Smith R.S. (2021). On Low-Rank Hankel Matrix Denoising. *The 19th IFAC Symposium on System Identification*.
- **Yin M.**, Iannelli A., Smith R.S. (2021). Maximum Likelihood Signal Matrix Model for Data-Driven Predictive Control. *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, PMLR 144:1004-1014.

- **Yin M.**, Iannelli A., Khosravi M., Parsi A., Smith R.S. (2020). Linear Time-Periodic System Identification with Grouped Atomic Norm Regularization. *IFAC World Congress*.
- Khosravi M.*, **Yin M.***, Iannelli A., Parsi A., Smith R.S. (2020). Low-Complexity Identification by Sparse Hyperparameter Estimation. *IFAC World Congress*.
- Khosravi M., Iannelli A., **Yin M.**, Parsi A., Smith R.S. (2020). Regularized System Identification: A Hierarchical Bayesian Approach. *IFAC World Congress*.
- Parsi A., Iannelli A., **Yin M.**, Khosravi M., Smith R.S. (2020). Robust Adaptive Model Predictive Control with Worst-Case Cost. *IFAC World Congress*.
- **Yin M.**, Chen Y., Lee K. H., Fu D. K., Tse Z. T. H., Kwok K. W. (2018). Dynamic Modeling and Characterization of the Core-XY Cartesian Motion System. *IEEE International Conference on Real-time Computing and Robotics*.

ETH Zurich
Automatic Control Laboratory
Physikstrasse 3
8092 Zurich, Switzerland