Data-Driven System Reduction and Identification from Input-Output Time-Domain Data with the Loewner Framework

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

von **M. Sc. Dimitrios S. Karachalios**

geb. am **11.07.1986** in Athen, Griechenland

genehmigt durch die Fakultät für Mathematik

der Otto-von-Guericke-Universität Magdeburg

Gutachter: **Prof. Dr. Athanasios C. Antoulas**

Prof. Dr. Peter Benner

Dr. habil. Charles Poussot-Vassal

Dr. Ion Victor Gosea

eingereicht am: **08.03.2023**

Verteidigung am: **18.07.2023**

PUBLICATIONS

Some of the results of this thesis have already been published or submitted.

Chapter 3 contains results of

- [102] Karachalios, Dimitrios S., Gosea, Ion Victor and Antoulas, Athanasios C.. Chapter 6: The Loewner framework for system identification and reduction. Volume 1 System- and Data-Driven Methods and Algorithms, edited by Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders and Luís Miguel Silveira, Berlin, Boston: De Gruyter, 2021, pp. 181-228. https://doi.org/10.1515/9783110498967-006

- [100] Karachalios, Dimitrios S., Gosea, Ion Victor and Antoulas, Athanasios C. Data-driven approximation methods applied to non-rational functions. Proc. Appl. Math. Mech., 2018: e201800368. https://doi.org/10.1002/pamm.201800368

Chapter 4 contains results of

- [107] Karachalios, D.S., Gosea, I.V., Kour K., and Antoulas, A.C., Bilinear realization from input-output data with neural networks. Accepted for publication in a Springer book series, The European Consortium for Mathematics in Industry. Preprint https://arxiv.org/abs/2208.10124

- [101] Karachalios, Dimitrios S., Gosea, Ion Victor and Antoulas, Athanasios C.. (2019), A bilinear identification-modeling framework from time domain data. Proc. Appl. Math. Mech., 2019: e201900246. https://doi.org/10.1002/pamm.201900246

- [104] Karachalios, Dimitrios S., Gosea, Ion Victor and Antoulas, Athanasios C. On Bilinear Time-Domain Identification and Reduction in the Loewner Framework, Springer International Publishing, Cham, 2021, pp. 3-30. iii, https://doi.org/10.1007/978-3-030-72983-7_1

- [73] Gosea, Ion Victor and Karachalios, Dimitrios S. and Antoulas, Athanasios C., On computing reduced-order bilinear models from time-domain data Proc. Appl. Math. Mech., 2021: e202100254. https://doi.org/10.1002/pamm.202100254

Chapter 5 contains results of

- [106] Karachalios, Dimitrios S., Gosea, Ion Victor, Gkimisis, Leonidas and Antoulas, Athanasios C. (2022), Data-driven quadratic modeling in the Loewner framework. Preprint https://arxiv.org/abs/2211.10635.
- [72] Gosea, Ion Victor, Karachalios, Dimitrios S., and Antoulas, Athanasios C. Learning reduced-order models of quadratic dynamical systems from input-output data, 2021 European Control Conference (ECC), 2021, pp. 1426-1431, https://doi:10.23919/ECC54610.2021.9654993.
- [105] Karachalios, Dimitrios S., Gosea, Ion Victor and Antoulas, Athanasios C. A framework for fitting quadratic-bilinear systems with applications to models of electrical circuits, IFAC-PapersOnLine, Volume 55, Issue 20, 2022, Pages 7-12, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2022.09.064

Chapter 6 contains results of

- [103] Karachalios, Dimitrios S., Gosea, Ion Victor and Antoulas, Athanasios C. The Loewner framework for nonlinear identification and reduction of Hammerstein cascaded dynamical systems. Proc. Appl. Math. Mech., 2020: e202000337. https://doi.org/10.1002/pamm.202000337

ABSTRACT

Consistent description of the physical world uses mathematics under the assumption of computability. The development of mathematics equipped the natural sciences with computational models capable of accurately describing each phenomenon at its scale of action while allowing the model's generalization without losing consistency with the observed local behavior. Accuracy in this content may have different meanings. The central concept of accuracy concerns the difference observed between experiment and theory. The experiment constitutes nature's response in which the answer may be subject to many different sources of disturbance (noise), inevitably leading to biases. Data-driven engineering science is the field that connects state-of-the-art theoretical methods with experimental results that guarantee reliable conclusions. Consequently, data-driven science aims to combine these two pillars (experiment and theory) consistently; in simple terms, the measurement must explain the theory and vice versa.

The idea of mimicking the human brain architecture (neurons) to a finite computational machinery for learning, predicting, and decision-making sparked the concept of artificial intelligence through a model-free computational environment fed with data (finite data and memory). Usually, these developments are referred to as machine learning techniques and aim to integrate data-driven engineering science after solving the model discovery problem (recent ultimate goal). Due to computing power and memory development, generating, storing, and processing data has become more accessible. Thus, modern high-performance computer environments reintroduced the idea of machine learning by allowing access to learning processes in vast databases for today's regimes. Consequently, with only a simple analysis, extracting difficult conclusions for decision-making has become possible and accurate (pattern recognition). Therefore, direct decision-making from data is advantageous but needs to mature enough to stand independently and replace data-driven engineering science when considering some guarantees on the outcome for safe predictions. It is a significantly different regime to decide on cinematic preferences or to understand an inaccurate translation relative to the precariousness that a plane can fly or a patient can breathe autonomously on mechanical ventilation. Failure in the above applications has an entirely different impact, and awareness should be raised.

Having posed some of the fundamental problems that science faces in this modern technological environment, this thesis aims to combine the advantages of the above research directions and contribute to solutions for the robust modeling of engineering processes respecting the mathematical formalism of the known physical laws for imposing reliability.

The main research directions apply to identifying dynamical systems that describe evolutionary phenomena with a mathematical formulation and computationally the efficient model order reduction of the discovered models under a finite assumed precision (known a priori). By including methods to reduce the order of the model under-recognition (from measurements), we arrive at applicable computational models where that adequately predict the behavior of the actual physical model (usually partially unknown), offering the possibility for simulation, design, control, and forecasting in the context of a reliable digital twin.

The main inspiration for this study comes from the assumption that, for experiments, every unbiased and undisturbed measurement is considered the absolute truth. We construct models using interpolation methods -"interpolating the truth"- such as the Loewner framework for identifying/inferring/building models based on these measurements as nature's proper response under some observable process of a potentially hidden perfect model. In addition, when the correct mathematical formalism of the model under construction has also been assumed, or the method allows its integration into the generally accepted class (embedding in a specific non-linear class), we can discover the physical law governing the phenomenon. The main result is the production of data-driven surrogate dynamical models whose mathematical formalism has matured and can connect with further theoretical properties for efficient simulation, design, and control, offering robustness and verification for accurate extrapolation.

ZUSAMMENFASSUNG

Die konsistente Beschreibung der physikalischen Welt mithilfe der Mathematik erfolgt unter Annahme der Berechenbarkeit. Die theoretische Weiterentwicklung der Mathematik stattete die Naturwissenschaften mit Rechenmodellen aus, die in der Lage sind, jedes Phänomen in seinem Wirkungsbereich präzise zu beschreiben und gleichzeitig eine Generalisierbarkeit des Modells zu ermöglichen, ohne die Übereinstimmung mit dem beobachteten lokalen Verhalten zu verlieren. Genauigkeit kann in diesem Zusammenhang verschiedene Bedeutungen haben. Der zentrale Begriff der Genauigkeit bezieht sich auf den Unterschied zwischen Experiment und Theorie. Das Experiment stellt die Reaktion der Natur dar, welche vielen verschiedenen Störgrößen (Rauschen) ausgesetzt sein kann, was unweigerlich zu Verzerrungen führt. Datengestützte Ingenieurwissenschaft verbindet modernste theoretische Methoden mit experimentellen Ergebnissen, die zuverlässige Schlussfolgerungen garantieren. Folglich zielt die datengesteuerte Wissenschaft darauf ab, diese beiden Säulen (Experiment und Theorie) kohärent miteinander zu verbinden; einfach ausgedrückt, die Messung muss die Theorie erklären und umgekehrt.

Die Idee, die Architektur des menschlichen Gehirns (Neuronen) mit einer endlichen Rechenmaschine für das Lernen, die Vorhersage und die Entscheidungsfindung zu imitieren, hat das Konzept der künstlichen Intelligenz durch eine modellfreie Rechenumgebung (endliche Datenmenge und Speicher) hervorgebracht. In der Regel werden diese Entwicklungen als maschinelles Lernen bezeichnet und zielen darauf ab, die datengesteuerte Ingenieurwissenschaft nach der Lösung des Modellfindungsproblems zu integrieren (jüngstes Endziel). Dank des Fortschreitens von Rechenleistung und Speicherkapazität ist die Erzeugung, Speicherung und Verarbeitung von Daten leichter zugänglich geworden. So haben moderne Hochleistungsrechenumgebungen die Idee des maschinellen Lernens wieder eingeführt, indem sie den Zugang von Lernprozessen zu riesigen Datenbanken für die heutigen Systeme ermöglichen. Folglich ist es möglich geworden, mit einer einfachen Analyse komplizierte Schlussfolgerungen für die Entscheidungsfindung zu ziehen (Mustererkennung). Daher ist die direkte Entscheidungsfindung auf der Grundlage von Daten vorteilhaft, muss aber noch so weit ausreifen, dass sie unabhängig ist und die datengesteuerte Ingenieurwissenschaft unter der Berücksichtigung von Garantien für sichere Vorhersagen ersetzen kann. Es hat eine ganz andere Dimension, über filmische Vorlieben zu entscheiden oder eine ungenaue Übersetzung zu verstehen, als ein Flugzeug zu fliegen oder einen Patienten künstlich zu beatmen. Ein Bewusstsein sollte vorhanden

sein, dass ein Versagen bei den genannten Anwendungen ganz andere Auswirkungen nach sich zieht.

Nachdem einige der grundlegenden Probleme, die die Wissenschaft im modernen technologischen Umfeld konfrontieren, erwähnt wurden, zielt diese Doktorarbeit darauf ab, die Vorteile der oben genannten Forschungsrichtungen zu kombinieren und dazu beizutragen, Lösungen für die robuste Modellierung von technischen Prozessen unter Beachtung des mathematischen Formalismus der bekannten physikalischen Gesetze zum Zwecke der Zuverlässigkeit zu gewährleisten. Die Forschungsschwerpunkte beziehen sich auf die Identifizierung dynamischer Systeme, welche evolutionäre Phänomene mit einer mathematischen Formulierung beschreiben sowie auf die rechnentechnisch effiziente Reduzierung der Modellordnung entdeckter Modelle unter einer endlichen angenommenen Genauigkeit (a priori bekannt). Durch die Einbeziehung von Methoden zur Verringerung der Ordnung des (aus Messungen) nicht erkannten Modells gelangen wir zu anwendbaren Rechenmodellen, die das Verhalten des tatsächlichen (in der Regel teilweise unbekannten) physikalischen Modells adäquat vorhersagen und damit die Möglichkeit zur Simulation, zum Entwurf, zur Steuerung und zur Vorhersage im Rahmen eines zuverlässigen digitalen Zwillings bieten.

Die Hauptinspiration für diese Studie geht von der Annahme aus, dass bei Experimenten jede unvoreingenommene und ungestörte Messung als absolute Wahrheit angesehen wird. Wir konstruieren Modelle mit Hilfe von Interpolationsmethoden - "Interpolation der Wahrheit" - wie z. B. dem Loewner-Rahmenwerk, um auf der Grundlage dieser Messungen die richtige Reaktion der Natur auf einen beobachtbaren Prozess eines potenziell verborgenen perfekten Modells zu ermitteln/zu bestimmen/zu konstruieren. Wenn außerdem der korrekte mathematische Formalismus des zu konstruierenden Modells angenommen wurde oder die Methode seine Integration in die allgemein akzeptierte Klasse (Einbettung in eine spezifische nichtlineare Klasse) ermöglicht, können wir das physikalische Gesetz entdecken, das das Phänomen erklärt. Das Hauptergebnis ist die Erstellung von datengesteuerten dynamischen Ersatzmodellen, deren mathematischer Formalismus ausgereift ist und mit weiteren theoretischen Eigenschaften für eine effiziente Simulation, Konstruktion und Steuerung verbunden werden kann, wobei Robustheit und Verifizierung für eine genaue Extrapolation geboten werden.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF SYMBOLS

$\mathbb{R}$, $\mathbb{C}$	fields of real and complex numbers
$\mathbb{C}_+$, $\mathbb{C}_-$	open right/open left complex half plane
$\mathbb{R}_+$, $\mathbb{R}_-$	strictly positive/negative real line
$\mathbb{R}^n$, $\mathbb{C}^n$	vector space of real/complex n-tuples
$\mathbb{R}^{m\times n}$, $\mathbb{C}^{m\times n}$	real/complex $m \times n$ matrices
$\|\xi\|$	absolute value of real or complex scalar
$\arg \xi$	argument of complex scalar
$\mathbb{D}$	$:= \{z \in \mathbb{C} : \|z\| < 1\}$, the open unit disc
j	imaginary unit ($j^2 = -1$)
$\mathrm{Re}(A)$, $\mathrm{Im}(A)$	real and imaginary part of a complex quantity $A = \mathrm{Re}(A) + \boldsymbol{\imath}\,\mathrm{Im}(A) \in \mathbb{C}^{n\times m}$
$\overline{A}$	$:= \mathrm{Re}(A) - j\,\mathrm{Im}(A)$, complex conjugate of $A \in \mathbb{C}^{n\times m}$
a_{ij}	the (i,j)-th entry of A
$A(i:j,:)$, $A(:,k:\ell)$	rows $i,\dots,j$ of A and columns $k,\dots,\ell$ of A
$A(i:j,k:\ell)$	rows $i,\dots,j$ of columns $k,\dots,\ell$ of A
A^{T}	the transpose of A
A^*	$:= (\overline{A})^{\mathrm{T}}$, the complex conjugate transpose
A^{-1}	the inverse of nonsingular A
$A^{-\mathrm{T}}$, A^{-*}	the inverse of A^{T}, A^*
I_n, $I_{n,r}$	identity matrix of dimension n, first r columns of I_n
$\mathbb{1}_r$	$:= (1,\dots,1)^{\mathrm{T}} \in \mathbb{R}^r$
$\Lambda(A)$, $\Lambda(A,M)$	spectrum of matrix A/matrix pair (A,M)
$\lambda_j(A)$, $\lambda_j(A,M)$	j-th eigenvalue of $A/(A,M)$
$\rho(A,M)$	$:= \max_j \|\lambda_j(A,M)\|$, spectral radius of (A,M)
$\sigma_{\max}(A)$	the largest singular value of A

$\mathrm{tr}(A)$	$:= \sum_{i=1}^{n} a_{ii}$, trace of A
$\lVert u\rVert_p$	$:= (\sum_{i=1}^{n} \lvert u_i\rvert^p)^{1/p}$ for $u \in \mathbb{C}^n$ and $1 \leq p < \infty$
$\lVert u\rVert_\infty$	$:= \max_i \lvert u_i\rvert$, the maximum norm of u
$\lVert A\rVert_p$	$:= \sup\{\lVert Au\rVert_p : \lVert u\rVert_p = 1\}$, subordinate matrix p-norm, $1 \leq p \leq \infty$
$\lVert A\rVert_{\mathrm{F}}$	$:= \sqrt{\sum_{i,j} \lvert a_{ij}\rvert^2} = \sqrt{\mathrm{tr}(A^*A)}$, the Frobenius norm of matrix $A \in \mathbb{C}^{m\times n}$
$\lVert u\rVert$, $\lVert A\rVert$	Euclidean vector or subordinate matrix norm $\lVert\cdot\rVert_2$
$\kappa_p(A)$	$:= \lVert A\rVert_p \lVert A^{-1}\rVert_p$, the p-norm condition number for nonsingular A
$\kappa(A)$	the 2-norm condition number for regular A
$A \succ (\succeq)0$, $A \prec (\preceq)0$	short form for A is self-adjoint positive/negative (semi)definite, also abbreviated by *s(s)pd* and *s(s)nd*
$A > (\geq)B$	$:\Leftrightarrow A - B > (\geq)0$, i.e., element-wise partial ordering: $(a_{ij} - b_{ij} > (\geq)0,\ \forall ij)$
$A \otimes B$	the Kronecker product of A and B
$\mathrm{vec}(A)$	vectorization operator applied to matrix A
$\partial_{x_j} f := \frac{\partial}{\partial x_j} f$	partial derivative with respect to x_j of f
$\partial_{x_j x_k} f = \frac{\partial^2}{\partial x_j \partial x_k} f$	$:= \partial_{x_j}\partial_{x_k} f$, second order partial derivative with respect to x_j and x_k of f
$\partial_{x_j}^2 f := \partial_{x_j x_j} f$	second order partial derivative with respect to x_j of f
$\dot{f} := \partial_t f := \frac{\partial}{\partial t} f$	the derivative with respect to time of f
$\ddot{f} := \frac{\partial^2}{\partial t^2} f$	second derivative with respect to time of f
$\partial^\alpha f := \partial_1^{\alpha_1} \dots \partial_n^{\alpha_n} f$	is the αth partial derivative of f, for the multi-index α
$\nabla f := (\partial_{x_1} f, \dots, \partial_{x_n} f)^{\mathrm{T}}$	the gradient of f
$\partial_\nu f := \nu \cdot \nabla f$	the derivative of f in direction of ν. In case of the outer normal simply the normal derivative of f.
$\Delta f := \sum_{i=1}^{n} \partial_{x_i}^2 f$	the *Laplacian operator* applied to f
$\nabla^2 f := \begin{bmatrix} \partial_{x_1}^2 f & \cdots & \partial_{x_1 x_n} f \\ \vdots & \ddots & \vdots \\ \partial_{x_n x_1} f & \cdots & \partial_{x_n}^2 f \end{bmatrix}$	the *Hessian matrix* of f

CHAPTER 1 INTRODUCTION

Contents

1.1. Motivation for data-driven identification and model order reduction

Mathematical modeling should be consistent with the physical laws when engineering applications are considered. Building surrogate models for robust simulation, design, and control is a worthwhile engineering task, and methods that can interpret or discover governing equations [40, 82] with guarantees [83] are essential and enhance modeling reliability. Modeling with partial differential equations (PDEs) allows continuous descriptions of the physical variables, and numerically, discrete approximations of PDEs result in systems of ordinary differential equations (ODEs). Even with the recent development of high-performance computing (HPC) environments, the resulting dynamical models still inherit high complexity that must be handled carefully. Therefore, approximation of large-scale dynamical systems [7] is pivotal for serving the scope of efficient simulation. The technique for reducing the complexity is known as model order reduction (MOR) [33, 30, 31]. There are many ways of reducing large-scale models, and each method is tailored to specific applications and goals for complexity reduction. A good distinction among methods concerns the accessibility or not of a high-fidelity model (intrusive or non-intrusive). For the intrusive case where a model is available, methods such as balanced truncation (BT) (see the recent survey [32]) and moment matching (MM) (with the recent survey [24] and the references therein) for constructing surrogate models of low order that approximate the original without losing much accuracy, offering an error bound (BT), and with a guarantee on stability (BT and some MM variants) were extensively used. Additional MOR methods for nonlinear systems were also developed (basically, by extending the linear counterpart of BT, MM, or others) [17, 8].

On the other hand, the ever-increasing availability of data, i.e., measurements related to the original model, initiate non-intrusive techniques such as Machine learning (ML) combined with model-based methods [40]. ML has demonstrated remarkable success in specific tasks, e.g., pattern recognition. The limitations of ML methods arise when the interpretation of the derived models is under consideration. Therefore, model-based data assimilation through MOR techniques such as the proper orthogonal decomposition (POD)[152], the dynamic mode decomposition. (DMD)[140, 133, 91], the operator inference (OpInf) [128, 28, 111] have become popular. In many cases, there may not be an accurate description of the original model (the large-scale dynamical system) but only at specific measurements (snapshots in the time domain, spectrum description, etc.). Therefore, one of the main challenges is the reliability of the information extracted from the data. The mentioned data-driven methods (OpInf, DMD) and others, such as the sparse identification of dynamical systems (SINDY)[41], use state-access snapshot measurements to achieve model discovery and recently to guarantee stability such as with OpInf in [84]. Towards the same aim of model discovery without using state-access measurements and building input invariant models, the Loewner framework constitutes a non-intrusive method that deals directly with i/o data (real-world measurements used in most of the applied sciences, e.g., frequency, velocity, voltage, charge, or concentration) able to identify linear and nonlinear systems and simultaneously to offer the opportunity for complexity reduction. One way to reduce the model complexity is to employ interpolation. Out of the many available existing methods, we mention those based on rational approximation. Here, we mention the Loewner framework [122], the vector fitting (VF) [88], and the AAA algorithm [123]. We refer the reader to the extensive analysis provided in [8] for more details on such methods.

The realization of linear models is introduced in [92] and has been extended further in [99]. For the nonlinear case, extensions to the realization algorithm (i.e., employing the subspace method) in the case of discrete-time bilinear control systems can be found in [95, 54, 44, 119] with the references within, and for linear parametric varying (LPV) systems in [148] when the scheduling signals can be measured. Other methods for data-driven system identification or reduction based on nonlinear autoregressive moving average with exogenous inputs (NARMAX) models can be found in [45] and in connection with Koopman operator and Wiener projection in [118]. Time discretization of semi-discretized in space nonlinear systems have disadvantages with the structure preservation for the resulting full-discretized model. A few schemes can preserve the structure, e.g., the forward Euler and for bilinear systems, but inherit conditional numerical stability, limiting the method to very short sampling times. A viable alternative is to devise methods that directly learn the continuous in-time operators without adding another discretization error due to the time mesh. The Fourier transform through the classical Nyquist theorem provides a way to transform discrete information into perfect continuous signal reconstruction from a finite spectrum if the correct sampling frequency has been considered.

The LF is a non-intrusive interpolatory MOR technique that identifies state-space systems for certain generalized nonlinear classes, particularly: bilinear systems [12], linear switched systems [74], linear parameter-varying systems [76], quadratic-bilinear systems

[65, 68, 9], and polynomial systems [25]. The aforementioned nonlinear variants of the Loewner framework construct efficient surrogate models from sampled data with direct numerical simulation (DNS) on the regular Volterra kernels derived from a prior accessible high-fidelity model. The challenging aspect of the regular multivariate Volterra kernels is that they cannot be inferred directly from a physical measurement setup when the underlying model is inaccessible and only a specific structure has been assumed (e.g., quadratic). Therefore, in a natural measurement environment (e.g., output in the time domain), the appropriate Volterra kernels we can measure are symmetric and can be derived with the growing exponential approach tailored to the probing method. The LF has already been extended to handle input-output data in the time domain for linear systems [127]. Consequently, we highlight the aim of this thesis that reflects the title once more next in Fig. 1.1 and in the following statement:

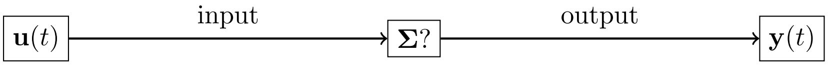


Figure 1.1.: A schematic of data-driven modeling, as the unknown to-be-discovered system Σ of low dimension lies between the accessible/measurable input-output data sequences through the time evolution.

The main novelty of this thesis is to devise methods that use or extend the interpolatory method known as the Loewner framework for nonlinear identification and reduction of dynamical systems from input-output time-domain data.

"An experiment is a question which science poses to Nature
and a measurement is the recording of Nature's answer
- **Max Planck**"

1.2. Dynamical systems and motivating examples

In this section, we introduce the types of dynamical systems investigated in this thesis and some motivational examples tailored to the corresponding mathematical formalism. Data-driven modeling of dynamical systems concerned with identifying the unknown system Σ in Fig. 1.1 with a state-space control system in various structures varying from linear to nonlinear as depends on the operators in Fig. 1.2. The reduction comes when a finite numerical precision has been assumed and usually is the trade-off between the complexity of the to-be-constructed model and accuracy performance.

When the identification task has been accomplished with all the validation procedures, we result in a "white box" modeling where further analysis can be obtained in many cases. For instance, stability analysis is handy if the identified model is linear or bilinear and easier to analyze when it comes to a state-space representation. When dynamics are measured around an operational point, linear (local) systems that perform well can be identified, but here someone should be aware that this is not white box modeling as for other inputs; the system can be driven to a different operational point, and the dynamical behavior can change dramatically. That is the main reason for nonlinear identification.

Next is a concise representation of state-space models starting from the linear time-invariant[1] systems and up to the quadratic state/input polynomial nonlinearity.

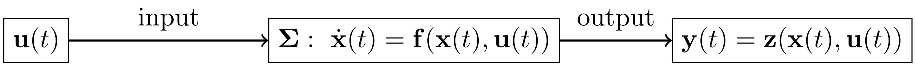

$\mathbf{u}(t)$
input
$\boldsymbol{\Sigma}: \ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$
output
$\mathbf{y}(t) = \mathbf{z}(\mathbf{x}(t), \mathbf{u}(t))$

Figure 1.2.: General state-space representation of the unknown system Σ with the time-invariant operators $\mathbf{f}, \mathbf{z}$.

Under specific assumptions, the unknown operators $\mathbf{f}, \mathbf{z}$ can be considered so to have the following state-space representations:

- **Linear control systems (Chapter 3)** The simplest case is that of a continuous linear time-invariant system

$$\boldsymbol{\Sigma}_l : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \ t \in [0, +\infty) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \ \mathbf{x}(0) = \mathbf{0}, \end{cases} \tag{1.1}$$

  where $\mathbf{E}, \ \mathbf{A} \in \mathbb{R}^{n\times n}, \ \mathbf{B} \in \mathbb{R}^{n\times m}, \ \mathbf{C} \in \mathbb{R}^{p\times n}$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$ are constant matrices due to the time-invariant principle.
  Examples: Mechanical systems, CD player, Euler Bernoulli beam, heat equation, function approximation.

- **Bilinear control systems (Chapter 4)** The next generalization from an LTI system is the class of bilinear control systems that belong to the nonlinear system class, as bilinear systems do not satisfy the superposition and scaling principles for general input construction. Bilinear systems are defined as:

$$\boldsymbol{\Sigma}_b : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)u(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \ \mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0}, \ t \geq 0. \end{cases} \tag{1.2}$$

  where $\mathbf{E}, \mathbf{A}, \mathbf{N} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$.
  Examples: The heat transfer model, the viscous Burgers' bilinear approximation, the heat exchanger.

- **Quadratic control systems (Chapter 5).**

$$\boldsymbol{\Sigma}_q : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{Q}\left(\mathbf{x}(t) \otimes \mathbf{x}(t)\right) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \ \mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0}, \ t \geq 0. \end{cases} \tag{1.3}$$

[1] A time-invariant (TI) system has a time-dependent system function that is not a direct function of time, e.g., $\mathbf{f}_1(\mathbf{x}(t), \mathbf{u}(t))$. For instance, a time-varying system is $\mathbf{f}_2(\mathbf{x}(t), \mathbf{u}(t), t)$.

where $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{Q} \in \mathbb{R}^{n\times n^2}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$.
Examples: The forced Lorenz attractor and the viscous Burgers' model with Robin boundary conditions.

- **Quadratic-bilinear control systems (Chapter 5).**

$$\boldsymbol{\Sigma}_{qb} : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{Q}\left(\mathbf{x}(t) \otimes \mathbf{x}(t)\right) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t),\ \mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0},\ t \geq 0. \end{cases} \tag{1.4}$$

where $\mathbf{E}, \mathbf{A}, \mathbf{N} \in \mathbb{R}^{n\times n}$, $\mathbf{Q} \in \mathbb{R}^{n\times n^2}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$.
Examples: The nonlinear RC ladder circuit.

- **Hammerstein-Wiener control system (Chapter 6)**

$$\boldsymbol{\Sigma}_{HW} : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{h}(\mathbf{u}(t)),\ t \in [0, +\infty) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{w}(\mathbf{x}(t)) + \mathbf{D}\mathbf{h}(\mathbf{u}(t)),\ \mathbf{x}(0) = \mathbf{0}, \end{cases} \tag{1.5}$$

where $\mathbf{E}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$ and $\mathbf{h}$, $\mathbf{w}$ are the Hammerstein and Wiener operators respectively. This cascaded system treats nonlinear effects from the input or output maps and assumes the internal dynamics to be linear.
Examples: Artificial examples to illustrate the methods of Hammerstein and Wiener system identification.

1.3. Outline of the thesis

In Chapter 2, we include some mathematical preliminaries from linear algebra and system theory. The experienced reader with linear algebra and system theory can go directly to the rest of the chapters.

In Chapter 3, we review the Loewner framework for identification and reduction in the linear case for the single-input, single-output (SISO), and multi-input multi-output (MIMO) cases with detailed introductory examples. We investigate descriptor system realizations with the Moore-Penrose pseudo inverse that extends the applicability of the Loewner framework to rectangular and singular systems. We introduce a new Loewner algorithm based on the CUR factorization that preserves the original data structure regarding sparsity and physical meaning. Also, analysis and some advances are given to technical issues such as data selection and splitting. We apply the aforementioned Loewner methods to various linear models and report the performance of reduction and accuracy. Approximation results are presented for infinite dimensional linear systems too. Applying the Laplace transformation to the PDE level, in most cases, we result in a non-rational transfer function. With the Loewner rational approximants, we could approximate the non-rational within the interpolation interval with high precision. The method is compared with other various methods, such as the adaptive Antoulas Anderson (AAA), the vector fitting (VF), and the iterative rational Krylov algorithm (IRKA) that can reach optimality in $\mathcal{H}_2$ norm.

In Chapter 4, we focus on identifying and reducing bilinear systems from time-domain data. We start with the continuous formulation of bilinear systems, and a method based on the Volterra series and the Loewner framework is proposed. Examples are presented for the identification and reduction goal of the proposed time bilinear Loewner. A comparison is offered with the developed bilinear Loewner that uses abstract frequency domain data as samples of higher Volterra kernels. To this effort, we propose combining the two bilinear Loewner frameworks by connecting different types of Volterra kernels. That was theoretically successful, but as detailed, it is numerically very challenging. Having the spectral information of the state evolution, we identify bilinear systems from state access information invariant of the input. This contrasts with other state access identification methods that are input-dependent. We continue with the discrete formulation of bilinear systems to achieve the same goals. We developed an algorithm for input-output bilinear identification from time-domain discrete data based on Isidori's work. We compared it with other various methods as the subspace bilinear identification methods. Moreover, we relax the number of simulations needed by utilizing machine learning techniques, particularly the neural network architecture, due to its power in learning input-output maps through the universal approximation theorem. The novelty is that we construct a reduced bilinear model from a single sequence of input-output data and the resulting system, which is an interpretable bilinear system suitable for white box modeling with ease of use and stability guarantees for robust control performance.

In Chapter 5, we extend the time domain bilinear Loewner method to quadratic control nonlinear systems and to quadratic-bilinear control systems for the reduction and identification scopes from time-domain input-output data. For the quadratic case, the novelty is that we achieve quadratic identification for low-order systems when more Volterra kernels are involved. In addition, we achieved global quadratic control identification for a quadratic system when bifurcated to its non-trivial equilibrium points that measurements held. An operator alignment problem had to be solved to identify the global quadratic system. Thus, another novelty to this effort is the solution algorithm that solves a constrained quadratic matrix equation by aligning two quadratic models without the access of the linear operator. Further, and more to the reduction performance, we test the proposed method for a larger scale example, and 98% reduction has been achieved with an overall normalized accuracy of more than 5 digits for measuring the velocity field of a flow. A similar method has been applied to a quadratic-bilinear application for a large-dimension nonlinear benchmark circuit.

In Chapter 6, we extend the Loewner framework to the Hammerstein cascaded system case. Also, some applications were provided for the Wiener cascaded system case.

CHAPTER 2 MATHEMATICAL PRELIMINARIES AND BASIC CONCEPTS

Contents

2.1. Linear algebra

2.1.1. The singular value decomposition

The SVD is arguably one of the most useful and commonly used tools in numerical linear algebra. It is listed as one of the main matrix decompositions and can be efficiently computed through various numerically stable algorithms. It is widely used for different high-dimensional reduction and approximation methods.

Any complex-valued matrix $\mathbf{A} \in \mathbb{C}^{n\times m}$ has a singular value decomposition given by $\mathbf{A} = \mathbf{Y}\Sigma\mathbf{X}^*$ where $\mathbf{Y} \in \mathbb{C}^{n\times n}$, $\mathbf{X} \in \mathbb{C}^{m\times m}$ are unitary matrices, i.e., $\mathbf{Y}^*\mathbf{Y} = \mathbf{I}_n$ and $\mathbf{X}^*\mathbf{X} = \mathbf{I}_m$. The left and right singular vectors are collected as columns of matrices $\mathbf{X}$, and $\mathbf{Y}$, respectively.

Additionally, the matrix $\Sigma \in \mathbb{C}^{n\times m}$ is defined as $\Sigma_{i,i} = \sigma_i$ and zero elsewhere. Here, the ordered non-negative scalars $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \sigma_n \geqslant 0$ are the singular values (for $n \leqslant m$).

It is assumed that matrix $\mathbf{A}$ has a low rank, i.e., $\text{rank}(\mathbf{A}) = r \leqslant n \leqslant m$. Let k be a positive integer so that $k < r$. The singular value decomposition of matrix $\mathbf{A}$ can be additively split as follows:

$$\mathbf{A} = \mathbf{Y}\cdot\Sigma\cdot\mathbf{X}^* = \underbrace{\begin{pmatrix}\mathbf{Y}_k & \mathbf{Y}_{n-k}\end{pmatrix}}_{n\times n}\cdot\underbrace{\begin{pmatrix}\mathbf{\Sigma}_k & \mathbf{0}_{k,m-k}\\ \mathbf{0}_{n-k,k} & \mathbf{\Sigma}_{n-k,m-k}\end{pmatrix}}_{n\times m}\cdot\underbrace{\begin{pmatrix}\mathbf{X}_k^*\\ \mathbf{X}_{m-k}^*\end{pmatrix}}_{m\times m} \tag{2.1}$$

$$= \underbrace{\mathbf{Y}_k\Sigma_k\mathbf{X}_k^*}_{:=\mathbf{A}_k} + \mathbf{Y}_{n-k}\mathbf{\Sigma}_{n-k,m-k}\mathbf{X}_{m-k}^* \tag{2.2}$$

where $\mathbf{Y}_k \in \mathbb{C}^{n\times k}$, $\mathbf{\Sigma}_k \in \mathbb{C}^{k\times k}$ and $\mathbf{X}_k \in \mathbb{R}^{m\times k}$. Note that the matrix

$$\mathbf{A}_k := \mathbf{Y}_k\Sigma_k\mathbf{X}_k^* \in \mathbb{C}^{m\times n}$$

can be written in terms of the truncated dyadic decomposition, i.e.,

$$\mathbf{A}_k = \sum_{i=1}^{k}\sigma_i y_i x_i^*,$$

where y_i and x_i are the i^{th} column of matrices $\mathbf{Y}$, and $\mathbf{X}$, respectively. A problem of interest is to approximate the original matrix $\mathbf{A}$ with a rank k matrix $\mathbf{T}$ so that the approximation error is minimal with respect to the 2-induced norm or the Frobenius norm. From the Schmidt-Eckart-Young-Mirsky theorem (see Theorem 3.6 in [7]), follows that (given $\sigma_k > \sigma_{k+1}$)

$$\min_{\mathbf{T}\in\mathbb{R}^{n\times m},\ \text{rank}(\mathbf{T})\leq k} \|\mathbf{A}-\mathbf{T}\|_2 = \sigma_{k+1}. \tag{2.3}$$

Moreover, it turns out that the unique solution to the minimization problem in (2.3) is given by $\mathbf{T} = \mathbf{A}_k$. If we replace the 2-induced norm with the Frobenius norm, it follows that

$$\min_{\mathbf{T}\in\mathbb{R}^{n\times m},\ \text{rank}(\mathbf{T})\leq k} \|\mathbf{A}-\mathbf{T}\|_F = \left(\sum_{i=k+1}^{n}\sigma_i^2\right)^{\frac{1}{2}}. \tag{2.4}$$

As before, the unique solution to the minimization problem in (2.4) is again given by $\mathbf{T} = \mathbf{A}_k$. For more details on the singular value decomposition (SVD), we refer the reader to [7], pages 31-41.

The advantage of the SVD is that it offers optimal low-rank solutions in both the 2-induced and Frobenius norms. At the same time, one disadvantage is given by the fact that the method (full SVD) has cubic complexity with respect to $\min(m, n)$ (in the classical set-up when applied to dense matrices). Considering this possible downside, we seek ways to circumvent the usage of the classical SVD and investigate other matrix decompositions. It is worth mentioning that SVD complexity can be faster than cubic for low-rank approximation with an iterative algorithm. A randomized version of SVD (r-SVD) will reveal this robust behavior in the latter.

2.1.2. A data-preserving matrix decomposition

A challenging aspect of data-driven approximation methods is the choice of a relevant and meaningful low-dimensional subset of the (usually large-scale) data set. In some cases, this subset can preserve relevant dynamics characteristics for the model described by the original data. In this framework, it is interesting to devise procedures to extract relevant information from large-scale measurements. The end goal is to construct reduced-order models suitable for control, design, and simulation tasks. Nowadays, the dimension of data sets for various applications can easily reach $\approx \mathcal{O}(10^6)$. In such cases, computing the SVD of large and full matrices becomes prohibitive.

One appealing alternative to SVD is the so-called CUR decomposition. As before, the goal is to approximate the original matrix $\mathbf{A} \in \mathbb{C}^{n\times m}$, by a product of three low-rank matrices $\hat{\mathbf{A}} = \mathbf{CUR}$. Here, the columns of the matrix $\mathbf{C} \in \mathbb{C}^{n\times c}$ represent a subset of the columns of $\mathbf{A}$ while the rows of the matrix $\mathbf{R} \in \mathbb{C}^{r\times m}$ form a subset of the rows of $\mathbf{A}$. Finally, the matrix $\mathbf{U} \in \mathbb{C}^{c\times r}$ is constructed such that the factorization $\hat{\mathbf{A}} = \mathbf{CUR}$ holds.

In this new setup, the left and right singular vectors appearing in the SVD are replaced by columns and rows of the initial matrix $\mathbf{A}$. Hence, the CUR factorization identifies essential rows and columns of a matrix $\mathbf{A}$. For more details on the CUR decomposition and some of its applications, we refer the reader to [62, 124, 63, 61, 121, 50, 146, 113].

The CUR factorization is hence an important tool for analyzing large-scale data sets, which offers the following advantages over SVD:

1. If the matrix $\mathbf{A}$ is sparse, then the matrices $\mathbf{C}$ and $\mathbf{R}$ are also sparse (unlike the matrices $\mathbf{X}$ and $\mathbf{Y}$ in the SVD approach).

2. The CUR factorization computes an approximation of A in terms of some of the rows and columns of $\mathbf{A}$. In contrast, the SVD computes approximants in terms of linear combinations of orthonormal bases generated by the rows and columns of $\mathbf{A}$.

3. Consider $\mathbf{A} \in \mathbb{R}^{m\times n}$, $m > n$. The complexity for computing the full SVD of $\mathbf{A}$ is $O(n^3)$ flops, using, for instance, the QR factorization, $O(mn^2)$ flops, using iterative methods as in ARPACK, and $O\left((m+n)k\right)$ flops per iteration, for approximate incremental methods where the k dominant singular triples are determined approximately (for details see [16]). On the other hand, the CUR factorization of order k requires $O(k^3 + k^2(m+n))$ flops per iteration (for details, see [113]).

Remark 2.1 (Implementation algorithms for CUR):
In Chapter 3, we will investigate algorithms that implement the CUR factorization based on the Cross Approximation and discrete empirical interpolation method (DEIM). $\diamondsuit$

2.1.3. The generalized inverse approach

Given the (rectangular) matrix $\mathbf{M} \in \mathbb{R}^{q\times k}$, the Moore-Penrose generalized inverse, denoted by $\mathbf{M}^{MP} \in \mathbb{R}^{k\times q}$, satisfies the following properties:

- $(a)\mathbf{M}\mathbf{M}^{MP}\mathbf{M} = \mathbf{M}$,

- (b) $\mathbf{M}^{MP}\mathbf{M}\mathbf{M}^{MP} = \mathbf{M}^{MP}$,
- (c) $\left[\mathbf{M}\mathbf{M}^{MP}\right]^T = \mathbf{M}\mathbf{M}^{MP}$,
- (d) $\left[\mathbf{M}^{MP}\mathbf{M}\right]^T = \mathbf{M}^{MP}\mathbf{M}$,

This generalized inverse always exists and is unique. We will be concerned with rectangular $q \times k$ polynomial matrices, which have an explicit (rank revealing) factorization as follows:

$$\mathbf{M} = \mathbf{X}\boldsymbol{\Delta}\mathbf{Y}^T,$$

where $\mathbf{X}$, $\boldsymbol{\Delta}$, $\mathbf{Y}$ have dimension $q \times n$, $n \times n$, $k \times n$, $n \leq q, k,$ and all have full rank n. In such cases, the Moore-Penrose generalized inverse is:

$$\mathbf{M}^{MP} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\boldsymbol{\Delta}^{-1}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T.$$

2.2. Linear systems theory

Time-domain formalism of a linear time-invariant dynamical system with m inputs, n internal variables (known as "states" in the case $\mathbf{E}$ is non-singular), and p outputs, is given as

$$\boldsymbol{\Sigma}_l : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \ t \in [0, +\infty) \\ \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \ \mathbf{x}(0) = \mathbf{0}, \end{cases} \tag{2.5}$$

where $\mathbf{E}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$ are constant matrices due to the time-invariant principle. Usually, the focus in considering systems of the form Eq. (2.5) treats them as defining maps from a vector space of inputs $\mathbf{u}(t) \in \mathcal{U}$ to a vector space of outputs $\mathbf{y}(t) \in \mathcal{Y}$, e.g., $\mathcal{F} : \mathcal{U} \to \mathcal{Y}$. Generally, the system is linear when the operator $\mathcal{F}$ satisfies the superposition and scaling principles. The description of the internal variable $\mathbf{x}(t)$ may or may not be interesting. Equation (2.5) describes a system that can be realized from the matrices ($\mathbf{E}$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$) without this representation to be unique. The invertible matrices $\mathbf{S}$, $\mathbf{T} \in \mathbb{R}^{n\times n}$ can produce a variety of different realizations ($\mathbf{SET}$, $\mathbf{SAT}$, $\mathbf{SB}$, $\mathbf{CT}$, $\mathbf{D}$) with identical input-output maps $\mathcal{F}$. That is important for the next description of linear time-invariant systems.

Concentrating in the ordinary differential equation Eq. (2.5), with a non-singular (e.g., invertible) $\mathbf{E}$ matrix, we have the following explicit solution

$$\begin{aligned}
&\dot{\mathbf{x}}(t) = \mathbf{E}^{-1}\mathbf{A}\mathbf{x}(t) + \mathbf{E}^{-1}\mathbf{B}u(t) \Rightarrow \frac{d}{dt}\left[e^{-\mathbf{E}^{-1}\mathbf{A}t}\mathbf{x}(t)\right] = e^{-\mathbf{E}^{-1}\mathbf{A}t}\mathbf{E}^{-1}\mathbf{B}u(t) \Rightarrow \\
&\quad \int_0^t \frac{d}{d\tau}\left[e^{-\mathbf{E}^{-1}\mathbf{A}\tau}\mathbf{x}(\tau)\right]d\tau = \int_0^t e^{-\mathbf{E}^{-1}\mathbf{A}\tau}\mathbf{E}^{-1}\mathbf{B}u(\tau)d\tau \Rightarrow \\
&e^{-\mathbf{E}^{-1}\mathbf{A}t}\mathbf{x}(t) - e^{-\mathbf{E}^{-1}\mathbf{A}\cdot 0}\mathbf{x}(0) = \int_0^t e^{-\mathbf{E}^{-1}\mathbf{A}\tau}\mathbf{E}^{-1}\mathbf{B}u(\tau)d\tau \Rightarrow \\
&\qquad\qquad \mathbf{x}(t) = e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{E}^{-1}\mathbf{A}(t-\tau)}\mathbf{E}^{-1}\mathbf{B}u(\tau)d\tau \Rightarrow (\text{with } \sigma = t - \tau) \\
&\qquad\qquad y(t) = \mathbf{C}e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{x}_0 + \int_0^t \underbrace{\mathbf{C}e^{\mathbf{E}^{-1}\mathbf{A}\sigma}\mathbf{E}^{-1}\mathbf{B}}_{h_1(\sigma)}u(t-\sigma)d\sigma + \mathbf{D}u(t).
\end{aligned} \tag{2.6}$$

The input-output solution of an LTI system with zero-initial conditions $\mathbf{x}_0 = \mathbf{0}$ and a zero feed-forward term $\mathbf{D} = 0$, results to the known convolution integral

$$y(t) = \int_0^t h_1(\sigma)u(t-\sigma)d\sigma = (h_1 * u)(t),\ t \geq 0. \tag{2.7}$$

In practice, even the explicit solution is given, and integration is quite challenging for large-scale systems. Therefore, many research directions try to circumvent these numerical issues by providing more robust and accurate algorithms. One way to avoid integration is to transform the problem with a nonlinear map. Such transformations usually belong to the class of spectral methods. Next, we introduce the Laplace transform, for which a particular case (integration over the imaginary axis $j\omega, \omega > 0$) is the well-known Fourier transform.

Frequency domain formalism of a linear time-invariant dynamical system in Eq. (2.5), with $\mathbf{u}(t)$ exponentially bounded, and under stability assumptions of $\boldsymbol{\sigma}_l$, the $x(t),\ y(t)$ will be exponentially bounded as well. Therefore, we may apply a Laplace transform $\mathcal{L}\{\cdot\}$ to Eq. (2.5) and solve for $\hat{\mathbf{y}}(s)$ (denotes the Laplace transform of $y(t)$) in terms of $\hat{\mathbf{u}}(s)$ (denotes the Laplace transform of $u(t)$).

$$\begin{cases} \mathcal{L}\left[\mathbf{E}\dot{\mathbf{x}}(t)\right] = \mathcal{L}\left[\mathbf{A}\mathbf{x}(t)\right] + \mathcal{L}\left[\mathbf{B}u(t)\right], \\ \mathcal{L}\left[y(t)\right] = \mathcal{L}\left[\mathbf{C}\mathbf{x}(t)\right] + \mathcal{L}\left[\mathbf{D}u(t)\right], \end{cases} \Rightarrow \begin{cases} s\mathbf{E}\hat{\mathbf{x}}(s) - \mathbf{E}\mathbf{x}_0 = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}\hat{\mathbf{u}}(s), \\ \hat{\mathbf{y}}(s) = \mathbf{C}\hat{\mathbf{x}}(s) + \mathbf{D}\hat{\mathbf{u}}(s). \end{cases} \tag{2.8}$$

Solving the algebraic equation Eq. (2.8) in terms of $\hat{\mathbf{x}}(s)$ (denotes the Laplace transform of $\mathbf{x}(t)$), and substituting to the output equation above, we conclude to

$$\hat{y}(s) = \left[\mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\right]\hat{u}(s). \tag{2.9}$$

This defines the transfer function of Eq. (2.5):

$$\mathbf{H}(s) := \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \tag{2.10}$$

whereby denoting the resolvent as

$$\boldsymbol{\Phi}(s) := (s\mathbf{E} - \mathbf{A})^{-1} \in \mathbb{C}^{n\times n}, \tag{2.11}$$

we can conclude to: $\mathbf{H}(s) = \mathbf{C}\boldsymbol{\Phi}(s)\mathbf{B} + \mathbf{D} \in \mathbb{C}^{p\times m}$. $\mathbf{H}(s)$ is a $p \times m$ matrix-valued rational function of the complex variable s and will uniquely determine the input-output map. As explained, the transfer function Eq. (2.10) remains the same under all equivalent descriptor realizations.

Remark 2.2 (The D-term):
No explicit $\mathbf{D}$ terms will be considered in the following system representations. The reason is that such terms can be incorporated in the remaining matrices of the realization, thus yielding what is known as a descriptor realization. Consider a rank-revealing factorization.

$$\mathbf{D} = \mathbf{D}_1\mathbf{D}_2 \quad \text{where} \quad \mathbf{D}_1 \in \mathbb{R}^{p\times\rho},\ \mathbf{D}_2 \in \mathbb{R}^{\rho\times m},$$

and $\rho = \operatorname{rank} \mathbf{D}$. It readily follows that:

$$\hat{\mathbf{E}} = \begin{bmatrix} \mathbf{E} & \\ & \mathbf{0}_{\rho\times\rho} \end{bmatrix},\ \hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \\ & -\mathbf{I}_{\rho} \end{bmatrix},\ \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{D}_2 \end{bmatrix},\ \hat{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \mathbf{D}_1 \end{bmatrix},$$

is a descriptor realization of the same system with no explicit **D**-term. The reason for not considering explicit **D** terms is that the Loewner framework yields descriptor realizations with the **D** term incorporated in the rest of the realization. For more details on considering a non-zero feed-forward **D** term, see Chapter 2 in [8]. ♢

2.2.1. Moments of a linear system

Given a matrix-valued function of time $\mathbf{h} : \mathbb{R} \to \mathbb{R}^{p\times m}$, its k^{th} moment is:

$$\eta_k = \int_0^\infty t^k \mathbf{h}(t)\, dt, \quad k = 0, 1, 2, \cdots . \tag{2.12}$$

If this function has a Laplace transform defined by $\mathbf{H}(s) = \mathcal{L}(\mathbf{h})(s) = \int_0^\infty \mathbf{h}(t)e^{-st}dt$, the k^{th} moment of $\mathbf{h}$ is, up to a sign, the k^{th} derivative of $\mathbf{H}$ evaluated at $s = 0$:

$$\eta_k = (-1)^k \left.\frac{d^k}{ds^k}\mathbf{H}(s)\right|_{s=0} \in \mathbb{R}^{p\times m}, \quad k = 0, 1, 2, \cdots . \tag{2.13}$$

In the sequel, we will also make use of a generalized notion of moments, namely the moments of $\mathbf{h}$ around an arbitrary point $s_0 \in \mathbb{C}$:

$$\eta_k(s_0) = \int_0^\infty t^k \mathbf{h}(t)e^{-s_0 t}dt. \tag{2.14}$$

These generalized moments turn out to be (up to a sign) the derivatives of $\mathbf{H}(s)$ evaluated at $s = s_0$:

$$\eta_k(s_0) = (-1)^k \left.\frac{d^k}{ds^k}\mathbf{H}(s)\right|_{s=s_0} \in \mathbb{R}^{p\times m},\ k = 0, 1, 2, \cdots . \tag{2.15}$$

In this context, assuming for simplicity that $\mathbf{E} = \mathbf{I}$, the moments of $\mathbf{h}$ at $s_0 \in \mathbb{C}$ are:

$$\eta_k(s_0) = -k\mathbf{C}(s_0\mathbf{I} - \mathbf{A})^{-(k+1)}\mathbf{B}, \quad k = 0, 1, 2, \cdots ,$$

provided that s_0 is not an eigenvalue of $\mathbf{A}$. Notice that the moments determine the coefficients of the Laurent series expansion of the transfer function $\mathbf{H}(s)$ in the neighborhood of $s_i \in \mathbb{C}$; in particular

$$\begin{aligned} \mathbf{H}(s) &= \mathbf{H}(s_0) + \mathbf{H}^{(1)}(s_0)\frac{(s-s_0)}{1!} + \cdots + \mathbf{H}^{(k)}(s_0)\frac{(s-s_0)^k}{k!} + \cdots \\ &= \eta_0(s_0) + \eta_1(s_0)\frac{(s-s_0)}{1!} + \cdots + \eta_k(s_0)\frac{(s-s_0)^k}{k!} + \cdots . \end{aligned}$$

2.2.2. Approximation by moment matching

Given $\mathbf{\Sigma} = (\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$, consider the expansion of the transfer function around s_i, $i = 1, \ldots, r$, as above. Approximation-by-moment matching consists of finding

$$\hat{\mathbf{\Sigma}} = (\hat{\mathbf{C}}, \hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}), \quad \hat{\mathbf{E}},\ \hat{\mathbf{A}} \in \mathbb{R}^{k \times k},\ \hat{\mathbf{B}} \in \mathbb{R}^{k \times m},\ \hat{\mathbf{C}} \in \mathbb{R}^{p \times k}, \tag{2.16}$$

such that the expansion of the transfer function

$$\hat{\mathbf{H}}(s) = \hat{\eta}_0(s_i) + \hat{\eta}_1(s_i)\frac{(s - s_i)}{1!} + \hat{\eta}_2(s_i)\frac{(s - s_i)^2}{2!} + \hat{\eta}_3(s_i)\frac{(s - s_i)^3}{3!} + \cdots,$$

for appropriate $s_i \in \mathbb{C}$, and $\ell_i, r \in \mathbb{N}$, satisfies:

$$\eta_j(s_i) = \hat{\eta}_j(s_i),\ j = 1, 2, \ldots, \ell_i \text{ and } i = 1, \ldots, r.$$

This problem can be seen as rational interpolation.

2.2.3. Rational interpolation by Petrov-Galerkin projection

A common way to reduce a system's complexity is through Petrov-Galerkin projections. Such projections are defined by means of two matrices $\mathbf{V}$, $\mathbf{W} \in \mathbb{R}^{n \times k}$, $k < n$, satisfying the condition that $\mathbf{W}^T\mathbf{V} \in \mathbb{R}^{k \times k}$ is invertible[1].

Definition 2.3 (Gelerkin & Petrov-Galerkin projectors):
Consider $\mathbf{v}_i,\ \mathbf{w}_i \in \mathbb{R}^n,\ i = 1, \ldots, k$, and let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$, $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_k] \in \mathbb{R}^{n \times k}$. The map defined by $\mathbf{\Pi}_1 = \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$, is an orthogonal projection onto the span of the columns of $\mathbf{V}$. If $\mathbf{W}^T\mathbf{V}$ is non-singular, $\mathbf{\Pi}_2 = \mathbf{V}(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T$, is an oblique projector onto the span of the columns of $\mathbf{V}$, along the columns of $\mathbf{W}$. $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ are usually referred to in the model reduction literature as **Galerkin** and **Petrov-Galerkin** projectors, respectively. ◊

Reducing the system $\mathbf{\Sigma} = (\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$ defined above, by means of a *Petrov-Galerkin projection*, we obtain the reduced system $\hat{\mathbf{\Sigma}} = \left(\hat{\mathbf{C}}, \hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}\right)$ with the reduced-order matrices given by:

$$\hat{\mathbf{C}} = \mathbf{C}\mathbf{V} \in \mathbb{R}^{p \times k},\ \hat{\mathbf{E}} = \mathbf{W}^T\mathbf{E}\mathbf{V},\ \hat{\mathbf{A}} = \mathbf{W}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{k \times k},\ \hat{\mathbf{B}} = \mathbf{W}^T\mathbf{B} \in \mathbb{R}^{k \times m}. \tag{2.17}$$

There are many ways of choosing Petrov-Galerkin projectors to achieve various goals. Here we will restrict our attention to interpolatory projections. Such projectors yield reduced models that match the moments of the original system. These moments are values of transfer functions at selected frequencies referred to as interpolation points.

Rational interpolation by projection was originally proposed in the work of Skelton and co-workers; see [151, 156, 155]. Contributions were also made by Grimme, Gallivan, and van Dooren [85, 57, 58].

[1] The notation $(\cdot)^T$ indicates transposition of $(\cdot)$, while the notation $(\cdot)^*$ indicates transposition of $(\cdot)$ followed by complex conjugation.

Suppose that we are given a system $\mathbf{\Sigma} = (\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$, assumed SISO (single-input single-output, i.e., $m = p = 1$) for simplicity. We wish to find lower-dimensional models $\hat{\mathbf{\Sigma}} = (\hat{\mathbf{C}}, \hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}})$, defined as in Eq. (2.17), $k < n$, such that $\hat{\mathbf{\Sigma}}$ approximates the original system in an appropriate way.

Consider k distinct points $s_j \in \mathbb{C}$. Define $\mathbf{V}$ as a generalized controllability matrix:

$$\mathbf{V} = \left[(s_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{B},\ \cdots\ ,\ (s_k\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\right] \in \mathbb{C}^{n\times k}, \tag{2.18}$$

and let $\mathbf{W}^*$ be any left inverse of $\mathbf{V}$. Then:

Proposition 2.4 (Rational interpolation via projection):
$\hat{\mathbf{\Sigma}}$ interpolates the transfer function of $\mathbf{\Sigma}$ at the points s_j, that is

$$\mathbf{H}(s_j) = \mathbf{C}(s_j\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} = \hat{\mathbf{C}}(s_j\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} = \hat{\mathbf{H}}(s_j),\ j = 1, \cdots, k.$$

Proof. Denoting by $\mathbf{e}_j = [0\cdots 0\ \underbrace{\ 1\ }_{j}\ 0\cdots\ 0]^T$, the j^{th} unit vector, we obtain the string of equalities below which leads to the desired result:

$$\begin{aligned}
\hat{\mathbf{C}}(s_j\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} &= \mathbf{CV}(s_j\mathbf{W}^*\mathbf{EV} - \mathbf{W}^*\mathbf{AV})^{-1}\mathbf{W}^*\mathbf{B} \\
&= \mathbf{CV}\left(\mathbf{W}^*(s_j\mathbf{E} - \mathbf{A})\mathbf{V}\right)^{-1}\mathbf{W}^*\mathbf{B} \\
&= \mathbf{CV}\left([* \ \cdots\ *\ \ \mathbf{W}^*\mathbf{B}\ \ *\ \cdots\ *]\right)^{-1}\mathbf{W}^*\mathbf{B} \qquad \square \\
&= \left[\mathbf{C}(s_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{B},\ \cdots\ ,\ \mathbf{C}(s_k\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\right]\mathbf{e}_j \\
&= \mathbf{C}(s_j\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}.
\end{aligned}$$

Next, we are concerned with matching the value of the transfer function at a given point $s_0 \in \mathbb{C}$, together with $k - 1$ derivatives. We define:

$$\mathbf{V} = \left[(s_0\mathbf{E} - \mathbf{A})^{-1}\mathbf{B},\ (s_0\mathbf{E} - \mathbf{A})^{-2}\mathbf{B},\ \cdots\ ,\ (s_0\mathbf{E} - \mathbf{A})^{-k}\mathbf{B}\right] \in \mathbb{C}^{n\times k}, \tag{2.19}$$

together with any left inverse $\mathbf{W}$. The following holds.

Proposition 2.5 (Higher moment rational interpolation):
$\hat{\mathbf{\Sigma}}$ interpolates the transfer function of $\mathbf{\Sigma}$ at s_0, together with $k - 1$ derivatives at the same point:

$$\frac{(-1)^j}{j!}\frac{d^j}{ds^j}\mathbf{H}(s)\Big|_{s=s_0} = \mathbf{C}(s_0\mathbf{E} - \mathbf{A})^{-(j+1)}\mathbf{B} = \hat{\mathbf{C}}(s_0\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-(j+1)}\hat{\mathbf{B}} = \frac{(-1)^j}{j!}\frac{d^j}{ds^j}\hat{\mathbf{H}}(s)\Big|_{s=s_0},$$

for $j = 0, 1, \cdots, k - 1$. ◊

Proof. Let $\mathbf{V}$ be as defined in (2.19), and $\mathbf{W}$ be such that $\mathbf{W}^T\mathbf{V} = \mathbf{I}_k$. It readily follows that the ℓ^{th} power of the projected matrix $s_0\hat{\mathbf{E}} - \hat{\mathbf{A}}$ satisfies:

$$(s_0\hat{\mathbf{E}} - \hat{\mathbf{A}})^{\ell} = [\ \underbrace{* \cdots *}_{\ell-1}\ \ \mathbf{W}^*\mathbf{B}\ \ \underbrace{* \cdots *}_{k-\ell}\ ].$$

Consequently $[\mathbf{W}^T(s_0\mathbf{E} - \mathbf{A})\mathbf{V}]^{-\ell}\mathbf{W}^T\mathbf{B} = \mathbf{e}_{\ell}$, which finally implies

$$\hat{\mathbf{C}}(s_0\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-\ell}\hat{\mathbf{B}} = \mathbf{CV}\left[\mathbf{W}^T(s_0\mathbf{E} - \mathbf{A})\mathbf{V}\right]^{-\ell}\mathbf{W}^T\mathbf{B} = \mathbf{CVe}_{\ell} = \mathbf{C}(s_0\mathbf{E} - \mathbf{A})^{-\ell}\mathbf{B},$$

for $\ell = 1, 2, \cdots, k$. □

Since any $\bar{\mathbf{V}}$ that spans the same column space as $\mathbf{V}$ achieves the same objective, projectors composed of combinations of the cases (2.18) and (2.19) achieve matching of an appropriate number of moments. To formalize this, we will make use of the following matrices:

$$\mathcal{R}_k(\mathbf{E}, \mathbf{A}, \mathbf{B}; \sigma) = \Big[(\sigma\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} \quad (\sigma\mathbf{E} - \mathbf{A})^{-2}\mathbf{B} \quad \cdots \quad (\sigma\mathbf{E} - \mathbf{A})^{-k}\mathbf{B}\Big].$$

Corollary 2.6 (Mixed moment matching):
(a) If $\mathbf{V}$ as defined above is replaced by $\bar{\mathbf{V}} = \mathbf{R}\mathbf{V}$, $\mathbf{R} \in \mathbb{R}^{k\times k}$, $det\,\mathbf{R} \neq 0$, and $\mathbf{W}$ by $\bar{\mathbf{W}} = \mathbf{R}^{-T}\mathbf{W}$, the same matching results hold true.

(b) Let $\mathbf{V}$ be such that $span\,col\,\mathbf{V} = span\,col\,[\mathcal{R}_{m_1}(\mathbf{E}, \mathbf{A}, \mathbf{B}; \sigma_1) \ \cdots \ \mathcal{R}_{m_\ell}(\mathbf{E}, \mathbf{A}, \mathbf{B}; \sigma_\ell)]$, and $\mathbf{W}$ be any left inverse of $\mathbf{V}$. The reduced system matches m_i moments at $\sigma_i \in \mathbb{C}$, $i = 1, \cdots, \ell$. ◊

2.2.4. Two-sided projections

The above results can be strengthened if the row span of the matrix $\mathbf{W}^T$ is chosen to match the row span of a generalized observability matrix. In such a case, twice as many moments can be matched with a reduced system of the same dimension. Here, in addition to points $s_1, \ldots, s_k$, and the associated Eq. (2.18), we are given k additional distinct points $s_{k+1}, \ldots, s_{2k}$. These points are used to define a generalized observability matrix:

$$\mathbf{W} = \left[(s_{k+1}\mathbf{E}^T - \mathbf{A}^T)^{-1}\mathbf{C}^T \quad \cdots \quad (s_{2k}\mathbf{E}^T - \mathbf{A}^T)^{-1}\mathbf{C}^T\right] \in \mathbb{C}^{n\times k}. \tag{2.20}$$

Proposition 2.7 (Double sided rational interpolation):
Assuming that $\mathbf{W}^T\mathbf{V}$ has full rank, the projected system $\hat{\mathbf{\Sigma}}$, interpolates the transfer function of $\mathbf{\Sigma}$ at the points s_i, $i = 1, \cdots, 2k$. ◊

Proof. The string of equalities that follows proves the desired result:

$$\begin{aligned}
\hat{\mathbf{C}}(s_i\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} &= \mathbf{C}\mathbf{V}\left(s_i\mathbf{W}^T\mathbf{E}\mathbf{V} - \mathbf{W}^T\mathbf{A}\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{B} \\
&= \mathbf{C}\mathbf{V}\left(\mathbf{W}^T(s_i\mathbf{E} - \mathbf{A})\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{B} \\
&= \mathbf{C}\mathbf{V}\left(\mathbf{W}^T[\cdots\ \mathbf{B}\ \cdots]\right)^{-1}\mathbf{W}^T\mathbf{B} \\
&= \mathbf{C}\mathbf{V}\mathbf{e}_i = \mathbf{C}(s_i\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}, \ \text{for } i = 1, \ldots, k, \qquad \square \\
&= \mathbf{C}\mathbf{V}\left(\begin{bmatrix} \vdots \\ \mathbf{C} \\ \vdots \end{bmatrix}\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{B} \\
&= \mathbf{e}_i^T\mathbf{W}^T\mathbf{B} = \mathbf{C}(s_i\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}, \ \text{for } i = k+1, \ldots, 2k.
\end{aligned}$$

As shown next, the projectors (see [145]) discussed in the previous section satisfy Sylvester equations.

Proposition 2.8 (Sylvester equations):
With $\mathbf{\Lambda} = \text{diag}\,[\lambda_1, \ \cdots, \ \lambda_k]$, $\mathbf{M} = \text{diag}\,[\mu_1, \ \cdots, \ \mu_q]$, where λ_i and μ_j are mutually distinct, $\mathbf{R} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^k$, and $\mathbf{L} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T \in \mathbb{R}^q$, the matrices $\mathbf{V}$ and $\mathbf{W}$ satisfy the Sylvester equations:

$$\mathbf{E}\mathbf{V}\mathbf{\Lambda} - \mathbf{A}\mathbf{V} = \mathbf{B}\mathbf{R} \quad \text{and} \quad \mathbf{M}\mathbf{W}^T\mathbf{E} - \mathbf{W}^T\mathbf{A} = \mathbf{L}\mathbf{C}. \tag{2.21}$$

◊

2.2.5. Interpolatory model order reduction

In MIMO (multi-input multi-output) systems, the moments are $p \times m$ matrices. So, in the case of rational matrix interpolation, the most appropriate way to proceed is to interpolate along specific directions; otherwise, the dimension will scale with the lengths of the input-output spaces. To avoid this dimensionality enlargement, a good way is to solve the so-called *tangential interpolation problem* (see, e.g., [13, 59, 52]).

More precisely, we are given a set of *input/output* response measurements specified by left driving frequencies $\{\mu_i\}_{i=1}^q \subset \mathbb{C}$, using left input or tangential directions: $\{\boldsymbol{\ell}_i\}_{i=1}^q \subset \mathbb{C}^p$, producing left responses: $\{\mathbf{v}_i\}_{i=1}^q \subset \mathbb{C}^m$, and right driving frequencies: $\{\lambda_i\}_{i=1}^k \subset \mathbb{C}$, using right input or tangential directions: $\{\mathbf{r}_i\}_{i=1}^k \subset \mathbb{C}^m$, producing right responses: $\{\mathbf{w}_i\}_{i=1}^k$. We are thus given the left data: $(\mu_j; \boldsymbol{\ell}_j{}^T, \mathbf{v}_j^T)$, $j = 1, \ldots, q$, and the right data: $(\lambda_i; \mathbf{r}_i, \mathbf{w}_i)$, $i = 1, \ldots, k$. The problem is to find a rational $p \times m$ matrix $\mathbf{H}(s)$, such that:

$$\mathbf{H}(\lambda_i)\mathbf{r}_i = \mathbf{w}_i, \ i = 1, \ldots, k, \quad \boldsymbol{\ell}_j^T\mathbf{H}(\mu_j) = \mathbf{v}_j^T, \ j = 1, \ldots, q. \tag{2.22}$$

The left data is rearranged compactly as:

$$\mathbf{M} = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_q \end{bmatrix} \in \mathbb{C}^{q \times q}, \quad \mathbf{L} = \begin{bmatrix} \boldsymbol{\ell}_1^T \\ \vdots \\ \boldsymbol{\ell}_q^T \end{bmatrix} \in \mathbb{C}^{q \times p}, \quad \mathbb{V} = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_q^T \end{bmatrix} \in \mathbb{C}^{q \times m}, \tag{2.23}$$

while the right data as:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_k \end{bmatrix} \in \mathbb{C}^{k \times k}, \quad \begin{array}{l} \mathbf{R} = [\mathbf{r}_1 \ \ \mathbf{r}_2 \ \ \cdots \ \ \mathbf{r}_k] \in \mathbb{C}^{m \times k}, \\ \\ \mathbb{W} = [\mathbf{w}_1 \ \ \mathbf{w}_2 \ \ \cdots \ \ \mathbf{w}_k] \in \mathbb{C}^{p \times k}. \end{array} \tag{2.24}$$

Interpolation points and tangential directions are determined by the problem or are selected to realize given model reduction goals.

$$\left.\begin{array}{l} \boldsymbol{\ell}_j^T\hat{\mathbf{H}}(\mu_j) = \boldsymbol{\ell}_j^T\mathbf{H}(\mu_j) \Rightarrow \boldsymbol{\ell}_j^T\hat{\mathbf{H}}(\mu_j) = \mathbf{v}_j, \ j = 1, \cdots, q, \\ \hat{\mathbf{H}}(\lambda_i)\mathbf{r}_i = \mathbf{H}(\lambda_i)\mathbf{r}_i \Rightarrow \hat{\mathbf{H}}(\lambda_i)\mathbf{r}_i = \mathbf{w}_i, \ i = 1, \cdots, k. \end{array}\right\} \tag{2.25}$$

For SISO systems, i.e., $m = p = 1$, left and right directions can be taken equal to one ($\boldsymbol{\ell}_j = 1, \mathbf{r}_i = 1$), and hence the conditions above become:

$$\left.\begin{array}{l} \hat{\mathbf{H}}(\mu_j) = \mathbf{H}(\mu_j) \Rightarrow \hat{\mathbf{H}}(\mu_j) = \mathbf{v}_j, \ j = 1, \cdots, q, \\ \hat{\mathbf{H}}(\lambda_i) = \mathbf{H}(\lambda_i) \Rightarrow \hat{\mathbf{H}}(\lambda_i) = \mathbf{w}_i, \ i = 1, \cdots, k. \end{array}\right\} \tag{2.26}$$

2.3. Nonlinear system theory with Volterra series representation

In this section, we introduce the Volterra series, which approximates nonlinear systems from a systems-theoretic perspective that keeps consistency with the concepts in the linear system theory unified and extends the transfer function in the nonlinear case appropriately. Here, we present the core concepts from Volterra's theory that will be the primary matter for supporting the aim of this thesis. For more details, the reader is advised to follow the books [137, 8, 139, 96].

Consider the following nonlinear system

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x})u(t), \\ \quad y(t) = \mathbf{m}(x), \ t \in [0, +\infty), \end{cases} \tag{2.27}$$

where $\mathbf{E},\ \mathbf{A} \in \mathbb{R}^{n\times n}$ are constant matrices; $\mathbf{x}(t) \in \mathbb{R}^n$ is the state, $\mathbf{u}(t) \in \mathbb{R}$ is an external forcing (scalar input), and $y(t) \in \mathbb{R}$ is a scalar output; and $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n,\ \mathbf{G} : \mathbb{R}^n \to \mathbb{R}$, and $\mathbf{m} : \mathbb{R}^n \to \mathbb{R}$. Extension to the MIMO systems is straightforward. Assume that the above operators $\mathbf{F}(\cdot),\ \mathbf{G}(\cdot)$, and $\mathbf{m}(\cdot)$ are analytic functions of their argument. In many applications, this is not a restrictive assumption. For instance, in most cases the map $\mathbf{m}$ is linear, thus $\mathbf{m}(\mathbf{x}(t)) = \mathbf{C}\mathbf{x}(t)$, where $\mathbf{C} \in \mathbb{R}^n$ is a constant vector. Moreover, for various flow problems, $\mathbf{F}$ is quadratic in $\mathbf{x}$, e.g., $\mathbf{x} \otimes \mathbf{x}$. Such systems are called linear-analytic systems since they are linear in the input, and the nonlinearities are analytic in $\mathbf{x}$. Now assume that the solution to unforced dynamics, i.e., $u(t) = 0$ in Eq. (2.27), exists in a time interval $[0, T]$. Then, there exists a scalar $\alpha > 0$ such that for all continuous bounded input function $u(t)$ with $\|u(t)\| < \alpha$, for $t \in [0, T]$, the output $y(t)$ can be written in terms of a Volterra series representation: There exist locally bounded, piecewise continuous functions $h_n : \mathbb{R}^n \to \mathbb{R}$ such that the next infinite series converges absolutely and uniformly on $[0, T]$.

$$y(t) = \sum_{n=1}^{N} y_n(t),\ y_n(t) = \int_0^t \cdots \int_0^t h_n(\tau_1, \ldots, \tau_n) \prod_{i=1}^{n} u(t - \tau_i) d\tau_i, \tag{2.28}$$

where $h_n(\tau_1, \ldots, \tau_n)$ is a real-valued function of $\tau_1, \ldots, \tau_n$ known as the nth-order time-domain Volterra kernel. After a multivariate Laplace transform to the time-domain kernels $h_n(\tau_1, \ldots, \tau_n)$, the nth-order generalized frequency response function (GFRF) is defined as

$$H_n(s_1, \ldots, s_n) = \int_0^\infty \cdots \int_0^\infty h_n(\tau_1, \ldots, \tau_n) e^{-\sum_{i=1}^{n} s_i \tau_i} d\tau_1 \cdots d\tau_n. \tag{2.29}$$

The above time-domain kernels h_n are considered causal. Thus, integration can be considered to the whole real line $\mathbb{R}$. The n^{th} Volterra operator is defined as:

$$V_n(u_1, u_2, ..., u_n) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, ..., \tau_n) \prod_{i=1}^{n} u_i(t - \tau_i) d\tau_i, \tag{2.30}$$

so that $y_n = V_n(u, u, ..., u)$ holds true.

Remark 2.9 (Homogeneity of the Volterra operator):
The map $u(t) \to y_n(t)$ is homogeneous of degree n, that is, $\alpha u \to \alpha^n y_n$, $\alpha \in \mathbb{C}$. Each Volterra kernel $h_n(t)$ determines a symmetric multi-linear operator. Small amplitudes (e.g., $|\alpha| < \epsilon$) will allow ordering the nonlinear terms in such a way that the terms having large powers of the amplitude (α^n) can be neglected. That is precisely the sense of approximating weakly nonlinear systems with the Volterra series. ◊

The mathematical formulations above are for general nonlinear systems. Therefore, one way to derive specific kernels is to assume some structure of the underlying system that will explain the measurements due to some knowledge of physics, such as inflow problems, e.g., Navier-Stokes equation, where the dynamics are described from quadratic models. Further, many engineering examples that are driven with general analytical nonlinearities, after applying lifting techniques [64], can be represented with polynomial nonlinearities (in the state variables) into the lifted space and the quadratic structure. As explained in [86], lifting strategies can result in polynomial systems of quadratic order in systems with ODEs or DAEs[2] where the non-invertible **E** in the latter case makes the problem quite challenging and is not the scope of this thesis.

2.3.1. A single-tone input

Consider the excitation of a system with an input consisting of two complex exponentials as in Eq. (2.31). Such inputs are typically used in electrochemical engineering applications as [129].

$$u(t) = A\cos(\omega t) = \left(\frac{A}{2}\right)e^{j\omega t} + \left(\frac{A}{2}\right)e^{-j\omega t}. \tag{2.31}$$

By using the above input in Eq. (2.28), we can derive the first Volterra term with $n = 1$ as:

$$\begin{aligned} y_1(t) &= \int_{-\infty}^{\infty} h_1(\tau_1)[u(t-\tau_1)]d\tau_1 \\ &= \frac{A}{2}e^{j\omega t}\underbrace{\int_{-\infty}^{\infty} h_1(\tau_1)e^{-j\omega\tau_1}d\tau_1}_{H_1(j\omega)} + \frac{A}{2}e^{-j\omega t}\underbrace{\int_{-\infty}^{\infty} h_1(\tau_1)e^{j\omega\tau_1}d\tau_1}_{H_1(-j\omega)} \Rightarrow \\ y_1(t) &= \frac{A}{2}\left(e^{j\omega t}H_1(j\omega) + e^{-j\omega t}H_1(-j\omega)\right). \end{aligned} \tag{2.32}$$

Similarly, for the 2nd term, we can derive the following:

$$y_2(t) = \left(\frac{A}{2}\right)^2\left[e^{2j\omega}H_2(j\omega,j\omega) + 2e^0H_2(j\omega,-j\omega) + e^{-2j\omega}H_2(-j\omega,-j\omega)\right]. \tag{2.33}$$

Remark 2.10 (Conjugate symmetry):
$H_2^*(j\omega,-j\omega) = H_2(-j\omega,j\omega),\ \forall\omega \in \mathbb{R}$. ◊

[2] DAEs: Differential algebraic equations

The input amplitude is A, the angular frequency is ω, the imaginary unit is j, the first order response function is $H_1(j\omega)$, and $H_n(j\omega,...,j\omega)$, for $n \geq 2$, are the higher-order frequency response functions (FRFs) or generalized frequency response functions (GFRFs). Then, the n^{th} Volterra term can be written as:

$$y_n(t) = \left(\frac{A}{2}\right)^n \sum_{p+q=n} {}^nC_q H_n^{p,q}(j\omega)e^{j\omega_{p,q}t}, \; \omega_{p,q} = (p-q)\omega. \tag{2.34}$$

Where the following notations have been used:

$$H_n^{p,q}(j\omega) = H_n(\underbrace{j\omega,...,j\omega}_{p-times};\underbrace{-j\omega,...,-j\omega}_{q-times}), \; \omega_{p,q} = (p-q)\omega, \; {}^nC_q = \frac{n!}{q!(n-q)!}. \tag{2.35}$$

Time-domain representation of harmonics

- **0^{th} harmonic-DC term**: Could be a shift to the output by a constant non-periodic term, known as DC term-y_{DC}[3].

$$y_{DC} = \sum_{n=1}^{\infty} \left(\frac{A}{2}\right)^{2n} {}^{2n}C_n H_{2n}^{n,n}(j\omega). \tag{2.36}$$

- **1^{st} harmonic**: By collecting all the terms with frequency ω we can derive the 1st harmonic as y_I.

$$y_I(t) = \left(\frac{A}{2}H_1(j\omega) + 3\left(\frac{A}{2}\right)^3 H_3(j\omega, j\omega, -j\omega) + ...\right)e^{j\omega t} + \text{c.t.} \tag{2.37}$$

  where c.t. stands for the "conjugate terms."

- **2^{nd} harmonic**: By collecting all the terms with frequency 2ω we can derive the 2nd harmonic as y_{II}.

$$y_{II}(t) = \left(\left(\frac{A}{2}\right)^2 H_2(j\omega, j\omega) + 4\left(\frac{A}{2}\right)^4 H_4(j\omega, j\omega, j\omega, -j\omega) + ...\right)e^{2j\omega} + c.t. \tag{2.38}$$

- **m^{th} harmonic**: The m^{th} harmonic in the *time-domain* can be computed by collecting the identical exponential power coefficients from Eq. (2.39) and by setting $p-q=m$, with $p=m+i-1$ and $q=i-1$ in Eq. (2.34). Hence, it follows that:

$$y_{m^{th}}(t) = \sum_{i=1}^{\infty} \left(\frac{A}{2}\right)^{m+2i-2} {}^{m+2i-2}C_{i-1} H_{m+2i-2}^{m+i-1,i-1}(j\omega)e^{jm\omega t}. \tag{2.39}$$

[3] DC: Direct Current

Frequency-domain representation of harmonics The m^{th} harmonic in the *frequency-domain* by applying single-sided Fourier transform in Eq. (2.39) is the following:

$$Y_{m^{th}}(jm\omega) = \sum_{i=1}^{\infty} \left(\frac{A}{2}\right)^{m+2i-2} {}^{m+2i-2}C_{i-1} H_{m+2i-2}^{m+i-1,i-1}(j\omega)\delta(jm\omega). \tag{2.40}$$

where $\delta(\cdot)$ is the Dirac delta distribution. When a single-tone input excites a nonlinear dynamical system, the steady-state frequency response is characterized by a spectrum with higher harmonics (as can be seen, for example, in Fig. 2.1). This behavior is not observed in the linear case, where only one harmonic appears at the input frequency.

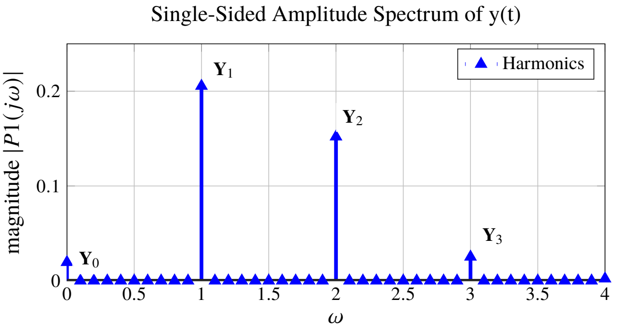


Figure 2.1.: An instance of the single-sided power spectrum with a singleton input. The underlying system is nonlinear; as a result, higher harmonics appeared with a DC (Direct Current, i.e., non-periodic) term.

2.3.2. The kernel separation method

One way to deduce Volterra kernels is by using interpolation. This problem is equivalent to that of estimating a polynomial with noisy coefficients. This interpolation scheme builds a linear system with an invertible Vandermonde matrix since the amplitudes are distinct and non-zero. The inverse of a Vandermonde matrix can be explicitly computed, and there are stable ways to solve these equations [37]. The recently proposed method presented in [39] solves the exponentially ill-condition problem of the Vandermonde matrix with Arnoldi orthogonalization. The frequency domain's m^{th} harmonic is derived using a (single-sided) Fourier transform. More precisely, the explicit formulation is as follows:

$$\begin{aligned} Y_{m^{th}}(jm\omega) &= \sum_{i=1}^{\infty} \underbrace{\left(\frac{A}{2}\right)^{m+2i-2} {}^{m+2i-2}C_{i-1}}_{\alpha^{m+2i-2}} H_{m+2i-2}^{m+i-1,i-1}(j\omega)\delta(jm\omega) \\ &= \sum_{i=1}^{\infty} \alpha^{m+2i-2} H_{m+2(i-1)}^{m+i-1,i-1}(j\omega)\delta(jm\omega). \end{aligned} \tag{2.41}$$

We simplify the notation to reveal the adaptive method to help us estimate the GFRFs up to a specific order. Next, write the linear system of equations that connects the harmonic information with the higher Volterra kernels as follows:

$$\underbrace{\begin{bmatrix} Y_0(0j\omega) \\ Y_1(1j\omega) \\ Y_2(2j\omega) \\ Y_3(3j\omega) \\ \vdots \\ Y_m(mj\omega) \end{bmatrix}}_{\mathbf{Y}_{(\alpha,\omega)}} = \left\{ \underbrace{\begin{bmatrix} \alpha^0 & \alpha^2 & \alpha^4 & \dots \\ \alpha^1 & \alpha^3 & \alpha^5 & \dots \\ \alpha^2 & \alpha^4 & \alpha^6 & \dots \\ \alpha^3 & \alpha^5 & \alpha^7 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \alpha^m & \alpha^{m+2} & \alpha^{m+4} & \dots \end{bmatrix}}_{\mathbf{M}_\alpha} \odot \underbrace{\begin{bmatrix} H_0^{0,0} & H_2^{1,1} & H_4^{2,2} & \dots \\ H_1^{1,0} & H_3^{2,1} & H_5^{3,2} & \dots \\ H_2^{2,0} & H_4^{3,1} & H_6^{4,2} & \dots \\ H_3^{3,0} & H_5^{4,1} & H_7^{5,2} & \dots \\ \vdots & \vdots & \vdots & \vdots \\ H_n^{n,0} & H_{n+2}^{n+1,1} & H_{n+4}^{n+2,2} & \dots \end{bmatrix}}_{\mathbf{P}_\omega} \right\} \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{\mathbf{e}_{n+1,1}}. \tag{2.42}$$

By introducing the Hadamard product notation[4] and by substituting the δ's with ones, we can compactly rewrite the above system in the following form:

$$\mathbf{Y}_{(\alpha,\omega)} = [\mathbf{M}_\alpha \odot \mathbf{P}_\omega] \cdot \mathbf{e}_{n+1,1}. \tag{2.43}$$

The above system offers the level of approximation we want to achieve. As we neglect higher-order Volterra kernels, the measurement set tends to be corrupted by noise. Note that the frequency response $\mathbf{Y}$ depends both on the amplitude and the frequency, while the right-hand side of Eq. (2.43) reveals the separation of the above quantities.

Remark 2.11:

Kernel separation and stage ℓ- approximation For a given system, the procedure involves exciting it with a single-tone input. By varying the driving frequency and amplitude, we can approximate the GFRFs by minimizing the (2-norm) of the remaining systems.

$$\mathbf{Y}_{m+1,\ell}(jm\omega, \alpha_\ell) = [\mathbf{M}_{m+1,\ell}(\alpha_\ell) \odot \mathbf{P}_{m+1,\ell}(jm\omega)] \cdot \mathbf{e}_{n+1,1}. \tag{2.44}$$

The m-"direction" gives us the threshold up to the specific harmonic we measure, while the ℓ-"direction" gives us the level of the kernel separation that we want to achieve. For instance, for the second stage approximation, we have $\ell = 2$ with $Y_m \approx 0,\ \forall m$ with $\ell = 2 < m = 3, 4, \ldots$. $\Diamond$

2.4. Representation of nonlinear systems with structured embeddings

2.4.1. Carleman linearization

Carleman in [43] introduced a technique capable of arbitrarily approximating any nonlinear system with analytical nonlinearities with the mildest representation of a nonlinear system that of bilinear control. Consider the general nonlinear system equivalent to Eq. (2.27) described as:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t),\ t \geq 0,\ \mathbf{x}(0) = \mathbf{0}. \end{cases} \tag{2.45}$$

[4] the Hadamard product is denoted with "$\odot$"; the matrix multiplication is performed element-wise.

The operators $\mathbf{f},\ \mathbf{g}:\mathbb{R}^n\to\mathbb{R}^n$ are supposed to be analytic in the state variable $\mathbf{x}$. The analysis shows that it is not restricted when $\mathbf{x}_0\neq\mathbf{0}$. In this case, the state shift $\hat{\mathbf{x}}(t)=\mathbf{x}(t)-\mathbf{x}_0$ results in the same structured system in Eq. (2.45).

The idea naturally arises: nonlinear systems with analytic operators $\mathbf{f},\ \mathbf{g}$ can be approximated with polynomial state representations similar to Taylor's approximation. The critical difference is that the polynomial degrees for the states scale with the Kronecker product $\otimes$. Denote the composition of k Kronecker products of the same vector $\mathbf{x}$ with itself as

$$\mathbf{x}^{(k)}(t):=\mathbf{x}(t)\otimes\mathbf{x}(t)\otimes\cdots\otimes\mathbf{x}(t)\in\mathbb{R}^{n^k}. \tag{2.46}$$

The (analytic) nonlinear operators $\mathbf{f}$ and $\mathbf{g}$ can be expanded as

$$\mathbf{f}(\mathbf{x})=\sum_{k=1}^{\infty}\mathbf{F}_k\mathbf{x}^{(k)},\quad \mathbf{g}(\mathbf{x})=\sum_{k=0}^{\infty}\mathbf{G}_k\mathbf{x}^{(k)}, \tag{2.47}$$

where $\mathbf{G}_0\in\mathbb{R}^{n\times 1},\ \mathbf{F}_j,\ \mathbf{G}_j\in\mathbb{R}^{n^j\times n^j},\ j\geq 1$. The system Eq. (2.45) has an equivalent representation to a state polynomial after substituting Eq. (2.47) with infinite dimension. The approximation involves the finite approximation of these operators by only N, similar to a Taylor approximation in which higher-order terms are neglected. Thus, it remains

$$\dot{\mathbf{x}}(t)=\sum_{k=1}^{N}\mathbf{F}_k\mathbf{x}^{(k)}(t)+\sum_{k=0}^{N-1}\mathbf{G}_k\mathbf{x}^{(k)}(t)\mathbf{u}(t)+res. \tag{2.48}$$

We write the derivative of $\mathbf{x}^{(2)}$ with respect to the time variable t, further, we use the Kronecker product property: $(\mathbf{A}\otimes\mathbf{B})(\mathbf{C}\otimes\mathbf{D})=(\mathbf{AC})\otimes(\mathbf{BD})$ and it remains:

$$\begin{aligned}\dot{\mathbf{x}}^{(2)}(t)&=\frac{d\mathbf{x}^{(2)}}{dt}=\dot{\mathbf{x}}(t)\otimes\mathbf{x}(t)+\mathbf{x}(t)\otimes\dot{\mathbf{x}}(t)=\\&=\left[\sum_{k=1}^{N}\mathbf{F}_k\mathbf{x}^{(k)}(t)+\sum_{k=0}^{N-1}\mathbf{G}_k\mathbf{x}^{(k)}(t)\mathbf{u}(t)\right]\otimes\mathbf{x}(t)+\mathbf{x}(t)\otimes[\sum_{k=1}^{N}\mathbf{F}_k\mathbf{x}^{(k)}(t)+\sum_{k=0}^{N-1}\mathbf{G}_k\mathbf{x}^{(k)}(t)\mathbf{u}(t)]\\&=\sum_{k=1}^{N}[\mathbf{F}_k\otimes\mathbf{I}_n+\mathbf{I}_n\otimes\mathbf{F}_k]\mathbf{x}^{(k+1)}(t)+\sum_{k=0}^{N-2}[\mathbf{G}_k\otimes\mathbf{I}_n+\mathbf{I}_n\otimes\mathbf{G}_k]\mathbf{x}^{(k+1)}(t)\mathbf{u}(t).\end{aligned} \tag{2.49}$$

Introduce the following notation that sums j terms, each containing $j-1$ Kronecker products, as follows (for $j\geq 2,\ k\geq 1$)

$$\mathbf{F}_{j,k}=\mathbf{F}_k(t)\otimes\mathbf{I}_n\otimes\cdots\otimes\mathbf{I}_n+\mathbf{I}_n\otimes\mathbf{F}_k(t)\otimes\mathbf{I}_n\otimes\cdots\otimes\mathbf{I}_n+\ldots+\mathbf{I}_n\otimes\cdots\otimes\mathbf{I}_n\otimes\mathbf{F}_k(t), \tag{2.50}$$

where we set $\mathbf{F}_{1,k}:=\mathbf{F}_k$. Similarly, define $\mathbf{G}_{j,k}$ for $j\geq 1,\ k\geq 0$. Now, write the time derivative of $\mathbf{x}^{(j)}(t)$ (for $j\geq 1$) in terms of the new defined matrices

$$\dot{\mathbf{x}}^{(j)}(t)=\sum_{k=1}^{N-j+1}\mathbf{F}_{j,k}\mathbf{x}^{(k+1)}(t)+\sum_{k=0}^{N-j}\mathbf{G}_{j,k}\mathbf{x}^{(k+1)}(t)\mathbf{u}(t). \tag{2.51}$$

Increase the dimension of the original state vector from n to $n^{(N)} = n + n^2 + \cdots + n^N = \frac{n^N - n}{n-1}$ introducing a new bilinear state variable (n is the dimension of the nonlinear variable $\mathbf{x}$)

$$\mathbf{x}^{\otimes}(t) = \begin{bmatrix} \mathbf{x}(t) & \mathbf{x}^2(t) & \cdots & \mathbf{x}^{(N)}(t) \end{bmatrix}^T \in \mathbb{R}^{n^{(N)}} \tag{2.52}$$

having introduced the above state vector, we obtain the following bilinear system representation

$$\begin{cases} \dot{\mathbf{x}}^{\otimes}(t) = \mathbf{A}^{\otimes}\mathbf{x}^{\otimes}(t) + \mathbf{N}^{\otimes}\mathbf{x}^{\otimes}(t)\mathbf{u}(t) + \mathbf{B}^{\otimes}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}^{\otimes}\mathbf{x}^{\otimes}(t),\ t \geq 0, \mathbf{x}_0^{\otimes}(0) = \mathbf{0}, \end{cases} \tag{2.53}$$

where the matrices have dimensions; $\mathbf{A}^{\otimes}$, $\mathbf{N}^{\otimes} \in \mathbb{R}^{n^{(N)} x n^{(N)}}$, $\mathbf{B}^{\otimes}$, $(\mathbf{C}^{\otimes})^T \in \mathbb{R}^{n^{(N)}}$ and are defined as:

$$\mathbf{A}^{\otimes} = \begin{bmatrix} \mathbf{F}_{1,1} & \mathbf{F}_{1,2} & \cdots & \mathbf{F}_{1,N} \\ \mathbf{0} & \mathbf{F}_{2,1} & \cdots & \mathbf{F}_{2,N-1} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{F}_{3,N-2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{F}_{N,1} \end{bmatrix}, \mathbf{N}^{\otimes} = \begin{bmatrix} \mathbf{G}_{1,1} & \mathbf{G}_{1,2} & \cdots & \mathbf{G}_{1,N-1} & \mathbf{0} \\ \mathbf{G}_{2,0} & \mathbf{G}_{2,1} & \cdots & \mathbf{G}_{2,N-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{3,0} & \cdots & \mathbf{G}_{3,N-3} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_{N,0} & \mathbf{0} \end{bmatrix}, \tag{2.54}$$
$$\mathbf{B}^{\otimes} = \begin{bmatrix} \mathbf{G}_{1,0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}^T, \mathbf{C}^{\otimes} = \begin{bmatrix} \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

Through this thesis, in many cases, especially in Chapter 4, we will use this structure for identification or reduction purposes. The Carleman procedure can generally be viewed as a linearization of the original nonlinear system Eq. (2.45) since it involves the approximation technique similar to Taylor series approximation that truncation of higher terms must be obtained. In this case, when the original system contains a control input $u(t)$, the resulting system is bilinear and has the form Eq. (2.53). For the non-controlled case, the resulting system is linear.

Remark 2.12 (Challenges of Carleman approximation):
The main challenges with this method are:

- It is an approximation of the original system.
- The dimension of the bilinear system scales exponentially w.r.t the original system and the order of the remaining polynomial terms and new states N. Usually, to keep the dimension tractable, in most cases, the terms are $N = 2$ (up to quadratic). $\diamondsuit$

Although the above remark may be seen as an inevitable task, this thesis aims to devise methods that can learn bilinear systems directly from input-output time-domain data that are also tailored with reduction techniques. Such a result will not affect the identified (low order) dimension, as we expect it to be relatively small in most cases. On the other hand, even for the intrusive methods that use the original large bilinear systems, depending on the application, model order reduction can also be performed, making the class of bilinear control systems a very appealing way of approximating general nonlinear systems.

2.4.2. Quadratization of nonlinear systems

The dynamics of many classical nonlinear PDEs or ODEs systems can be expressed after lifting in terms of polynomial nonlinearities without any approximation error. Examples include the Chafee-infante, FitzHugh-Nagumo, Burgers', Oseen, Stokes, RC ladder, or Navier-Stokes equations. This is performed by using specifically tailored lifting transformations. In particular, the method of lifting works as follows; Auxiliary variables and equations are artificially introduced to reformulate the equations into the structure. This allows the application of conventional MOR methods to more general nonlinear systems. Specific lifting transformations have already been discussed in the past. Kerner and McCormick have also introduced lifting techniques for representing nonlinear systems to exact polynomials. Most recently, algorithms for lifting to quadratic and for many engineering examples tailored with the current study have been introduced in [86, 114, 38], and in the recent study [64].

Examples above, e.g., the viscous Burgers' equation or the RC ladder nonlinear circuit, will be analyzed directly in the lifted version, usually bilinear or quadratic-bilinear. Some more examples will be introduced toward the end of the thesis and in chapter Chapter 6, where the analytical steps of bringing a general nonlinear system with analytic nonlinearities to the quadratic form will be analyzed.

Remark 2.13 (Advantages and challenges):
The main idea of quadratization via lifting strategies has two main advantages:

- Transforms the original nonlinear system equivalently to the quadratic control system form.
- Many methods developed for the class of quadratic systems can be applied directly and subsequently to any nonlinear system that can be transformed into the quadratic form.

Although the two above characteristics seem fairly powerful and, in many cases, provide good results, it must be considered that numerical stiffness is introduced due to lifting, making the simulation performance lower and sometimes quite challenging. Further, lifting inserts auxiliary variables; thus, it increases the dimension of the resulting system without guaranteeing a convergent procedure and that only finite terms need to be introduced. Moreover, by not keeping the structure of the original nonlinear system, the linear operator seems to contain huge sparse blocks with only a few non-zero eigenvalues. Most model reduction methods respect the splitting as in Eq. (2.27) with a significant matrix $\mathbf{A}$. If the linear operator diminishes, such cases have only a few results. ♢

Example 2.1 (The forced Van der Pol oscillator):
In dynamics, the Van der Pol oscillator is a non-conservative oscillator with non-linear damping. It evolves in time according to the second-order differential equation:

$$\frac{d^2y}{dt^2} - \mu(1-y^2)\frac{dy}{dt} + y = u(t) \tag{2.55}$$

$y(t)$ is the position as a function of time, $u(t)$ is the external force, and μ is a scalar parameter indicating the damping's nonlinearity and strength.

The second-order differential Eq. (2.55) can be written as a first-order by introducing the following states; $x_1(t) := y(t),\ x_2(t) := \frac{dy}{dt} = \dot{y}(t)$. Obviously, $\dot{x}_2(t) = \ddot{y}(t)$. Thus, we can write:

$$\mathbf{\Sigma}_{\text{cubic}} \begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = \mu(1 - x_1^2(t))x_2(t) - x_1(t) + u(t). \end{cases} \tag{2.56}$$

The polynomial system in Eq. (2.56) has cubic degree due to the term $x_1^2(t)x_2(t)$.

Lifting to quadratic. Quadratization procedure introduces auxiliary variables by increasing the state dimension. For instance, we can introduce the $x_3(t) := x_1^2(t)$ as a third state. We must differentiate first for augmenting this state with the original 1st order system.

$$\dot{x}_3(t) = \frac{dx_3}{dt} = 2x_1(t)\dot{x}_1(t) = 2x_1(t)x_2(t).$$

The remaining lifted system in quadratic form is

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = -x_1(t) + \mu x_2(t) - \mu x_3(t)x_2(t) + u(t), \\ \dot{x}_3(t) = 2x_1(t)x_2(t), \end{cases} \tag{2.57}$$

with operators

$$\mathbf{A}_q = \begin{bmatrix} 0 & 1 & 0 \\ -1 & \mu & 0 \\ 0 & 0 & 0 \end{bmatrix},\ \mathbf{Q}_q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mu/2 & 0 & -\mu/2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},\ \mathbf{B}_q = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{2.58}$$

The subscript "q" indicates that the resulting system's operator comes from a polynomial state system with a maximum degree equal to two (quadratic nonlinearities). If we choose to observe the position $x_1(t) = y(t)$, then, the output vector is $\mathbf{C}_q = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, and with state vector $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$, the following quadratic control system is derived:

$$\mathbf{\Sigma}_{\text{quadratic}} : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}_q\mathbf{x}(t) + \mathbf{Q}_q(\mathbf{x}(t) \otimes \mathbf{x}(t)) + \mathbf{B}_q u(t), \\ y(t) = \mathbf{C}_q\mathbf{x}(t). \end{cases} \tag{2.59}$$

Remark 2.14 (Quadratization of the forced Van der Pol oscillator):
Through this small-scale example, we want to include some remarks.

- The two systems in Eq. (2.56) and Eq. (2.59), although are different in the internal description (different state dimension and polynomial degree), they have the same external description. Thus, the two input-output maps are equivalent.

- The main reason for lifting to quadratic systems is that many model order reduction methods have been extended to this class of nonlinear systems (e.g., the Loewner framework). Thus, they can directly apply when all the operators are accessible from the high-fidelity model and not in a purely data-driven way.

- On the other hand, the different internal descriptions can have different numerical performances due to the dimensionality scaling. Numerical stiffness has been observed for lifted systems from a higher polynomial degree to quadratic. More details on the appropriate polynomial degree in [25].

- For the non-intrusive case and towards the methods we aim to develop in this thesis, keeping the original polynomial structure is beneficial as the linear operator is theoretically minimal, and the number of eigenvalues matches the state dimension of the system under some controllability/observability conditions. Thus, for our approaches in this thesis, for a system like the forced Van der Pol oscillator, it is better to infer the input-output dynamics from a cubic system (original) instead of a lifted quadratic where the linear operator loses its strength of giving the correct dimensionality order, e.g., $\texttt{eig}(\mathbf{A}_q) = \mu/2 \pm \sqrt{\mu^2 - 4}/2$ (two non-zero eigenvalues but the order is 3). $\Diamond$

CHAPTER 3

THE LOEWNER FRAMEWORK FOR LINEAR SYSTEMS

Contents

3.1. Introduction

One of the main approaches to the model reduction of linear dynamical systems is employing interpolation. This approach seeks reduced models whose transfer function matches the original system at selected interpolation points. Data-driven methods constitute an important particular class of modeling methods. We start with an account of the Loewner framework in the linear case [122]. It constructs models from given data straightforwardly. An important attribute is that it provides a trade-off between the accuracy of fit and the complexity of the model. We compare this approach with other approximation methods and apply it to test cases. One of the case studies to which we apply the aforementioned methods is defined by the inverse of the Bessel function. We then turn our attention to the approximation of an Euler-Bernoulli beam model with Rayleigh damping.

Further case studies include approximating two real-valued functions with specific difficulties (discontinuity, sharp peaks) and other nontrivial approximation tasks. One computational tool is the SVD; its complexity is cubic in the number of data points. For large data sets, the CUR factorization is a viable alternative. Note that its complexity is also cubic, but concerning the dimension of the reduced-order model (ROM). Another option is to use stochastic procedures such as randomized singular value decomposition (r-SVD) [89].

A complex problem facing computational linear algebra is that of modeling big data. The problem mainly involves constructing reduced complex systems from input/output data. This contribution focuses on reduction via interpolation. The Loewner framework is a data-driven approach that can construct low-order models from measurements. It can be applied to frequency and time-domain data [127]. In this chapter, we concentrate on frequency domain data; later, we will connect it naturally with the time domain. The Loewner framework will be implemented using (a) the SVD (singular value decomposition), (b) the CUR factorization, and (c) randomized SVD (r-SVD). Its performance will be compared with that of the recently developed AAA algorithm see [123], the Vector Fitting approach [88, 52], and the IRKA algorithm [18]. Chapter 3 is composed by three subsections. Section 3.2 covers the fundamentals of the Loewner framework, starting from a left and right interpolatory reduction in connection with moment matching. It concludes (a) by describing an interpolation property satisfied by reduced systems and (b) by making the procedure of obtaining real reduced models (despite complex interpolation points and values) explicit. Next, the description of two algorithms, namely, Loewner-SISO and Loewner-MIMO, is given. Finally, numerical examples are presented, and the role of generalized inverses Section 2.1.3 is outlined. Describes methods for implementing the Loewner reduction, namely the SVD, the CUR factorization, and the role of splitting the interpolation point in left and right sets. Illustrates the main features of the Loewner approach utilizing seven case studies, namely, (a) the CD player, (b) an oscillating function, (c) the inverse of a Bessel function, (d) an Euler-Bernoulli beam, (e) a heat equation, (f) a function with two sharp peaks, and (g) the sign function.

3.2. The Loewner framework

The Loewner framework has attracted the increased attention of researchers from various fields of applied mathematics and control engineering in the last 20 years. Consequently, a fair number of contributions that are now available deal with various aspects of further extending the framework and its application to different test cases. Below we provide an account of some of the works related to or inspired by the "Loewner framework."

Table 3.1.: A collection of contributions related to the Loewner Framework

Original paper [122] & tutorial paper [13]	Chapters 4 and 7 in the book [8]
Extension to	Application to
LPV & parametrized linear systems [6, 93, 76]	modeling multi-port linear systems [115]
bilinear systems [12, 73, 104, 101]	preserving the stability of the ROM [66, 77]
quadratic systems [69, 65, 72, 48, 106]	the Burger's equations [10, 48, 106]
quadratic-bilinear systems [69, 105]	the Oseen equations [11]
linear switched systems [75]	preserving the structure of DAE systems [80]
polynomial systems [26, 5]	systems with delay [141, 71]
modeling from noisy data [51, 117, 109]	approximating functions [100, 123]
modeling from time-domain data [127, 104, 106]	singular/rectangular systems [4]
one-sided Loewner framework [70]	genes oscillations [14] & biological rhythms [157]
data-driven control [79, 78, 110]	generic BizJet gust load alleviation [132]
port Hamiltonian systems [29, 94]	flutter stability with true damping [134]
second order systems [131]	batteries [138], electrochemistry [81, 126, 147]
Perspective based on duality and application to bilinear differential [135, 136]	Interpretation based on interconnection and application to LPV/LTV systems [142, 143, 76]

3.2.1. Rational interpolation and the Loewner matrix

In this section, we deal with the Lagrangian rational interpolation that is numerically proven to be more stable than the classical polynomial interpolation. There are certain bases, e.g., Chebynshev base, for which the polynomial interpolation is almost optimal, and for further details [149] is of relevance. Still, here we will be concerned with the rational approximation problem, which is more general and tailored to the system theory as the transfer function belongs to the class of rational functions. The general rational function of degree (n, d) over the complex plane $\mathbb{C}$ is defined as:

$$r(s) := \frac{\beta_n s^n + \beta_{n-1} s^{n-1} + \cdots + \beta_1 s + \beta_0}{\alpha_n s^d + \alpha_{d-1} s^{d-1} + \cdots + \alpha_1 s + \alpha_0},\ s \in \mathbb{C}, \tag{3.1}$$

where (n, d) tells the order of the numerator and denominator, respectively. Moreover, $\beta_n,\ \alpha_m \neq 0$. The above rational function Eq. (3.1) covers the polynomial function, thus making rational interpolation more general. In particular, polynomials have poles "at infinity." In contrast, for the rational function case, singularities can be finite, and Eq. (3.1) is the right structure for approximating functions with singular points, i.e., roots of the denominator. For simplicity, let's take equal polynomial degrees for the numerator and denominator; thus, $d = n$.

The unknown coefficients at first glance seem $2n+2$, but after division with the nonzero $\alpha_n \neq 0$, we get the normalized version of the following rational function of degree (n, n)

$$r(s) := \frac{\hat{\beta}_n s^n + \hat{\beta}_{n-1} s^{n-1} + \cdots + \hat{\beta}_1 s + \hat{\beta}_0}{\hat{\alpha}_n s^n + \hat{\alpha}_{n-1} s^{n-1} + \cdots + \hat{\alpha}_1 s + \hat{\alpha}_0}, \ s \in \mathbb{C}, \ \hat{\alpha}_n = 1. \tag{3.2}$$

To compute the rational interpolant of order n, one needs to determine $2n+1$ coefficients so that the following interpolation conditions are satisfied

$$r(s_i) = f_i, \ i = 1, 2, \ldots, 2n+1, \quad \text{where} \quad r(s) = \frac{\sum_{k=0}^{n} \hat{\beta}_k s^k}{\sum_{k=0}^{n} \hat{\alpha}_k s^k}. \tag{3.3}$$

Firstly, partition the set of interpolation nodes into two disjoint sets

$$\{s_1, s_2, \ldots, s_{2n+1}\} = \{\mu_1, \mu_2, \ldots, \mu_n\} \cup \{\lambda_1, \lambda_2, \ldots, \lambda_n, \lambda_{n+1}\}.$$

Secondly, partition the set of points into the analogs disjoint sets from the nodes.

$$\{f_1, f_2, \ldots, f_{2n+1}\} = \{\upsilon_1, \upsilon_2, \ldots, \upsilon_n\} \cup \{\omega_1, \omega_2, \ldots, \omega_n, \omega_{n+1}\}.$$

The most crucial numerical handling is to avoid using the monomial base expansion of the Eq. (3.3) as it will lead to an ill-conditioned Vandermonde matrix, but instead to introduce the Lagrange basis $\ell(s)$ and to rewrite the rational interpolation problem by using the barycentric formula.

$$r(s_i) = f_i, \ i = 1, 2, \ldots, 2n+1, \quad \text{where} \quad r(s) = \frac{\sum_{i=1}^{n+1} b_i \ell_i(s)}{\sum_{i=1}^{n+1} a_i \ell_i(s)}, \tag{3.4}$$

where the Lagrange polynomials are given $\ell_i(s) = \prod_{k=1,k\neq i}^{n+1}(s - \lambda_k), \ i = 1, 2, \ldots, n+1$. Evaluating Eq. (3.4) at the nodes λ_i, it follows that $r(\lambda_i) = b_i/a_i = \omega_i \Rightarrow b_i = \alpha_i \omega_i$. By dividing both the numerator and denominator in Eq. (3.4) with the polynomial $\ell(s) = \prod_{k=1}^{n+1}(s - \lambda_k)$, it follows that the barycentric formula is rewritten as

$$r(s) = \frac{\sum_{i=1}^{n+1} \frac{b_i}{s-\lambda_i}}{\sum_{i=1}^{n+1} \frac{\alpha_i}{s-\lambda_i}} = \frac{\sum_{i=1}^{n+1} \frac{a_i \omega_i}{s-\lambda_i}}{\sum_{i=1}^{n+1} \frac{\alpha_i}{s-\lambda_i}}, \tag{3.5}$$

and to recover the rational function r that interpolates all the data, we need to specify the $n+1$ unknowns coefficients (weights) $a_i, \ i = 1, \ldots, n+1$. The reduction from $2n+1$ unknown to only $n+1$ has to do with the fact we have already enforced $r(\lambda_i) = \omega_i$. It remains to enforce the remaining interpolation conditions

$$r(\mu_j) = \upsilon_j \Leftrightarrow \frac{\sum_{i=1}^{n+1} \frac{a_i \omega_i}{\mu_j - \lambda_i}}{\sum_{i=1}^{n+1} \frac{\alpha_i}{\mu_j - \lambda_i}} = \upsilon_j, \ \forall 1 \leq j \leq n \Leftrightarrow \sum_{j=1}^{n} \sum_{i=1}^{n+1} \frac{\upsilon_j - \omega_i}{\mu_j - \lambda_i} \alpha_i = 0,$$

where in matrix form is

$$\mathbb{L} = \begin{bmatrix} \frac{\upsilon_1 - \omega_1}{\mu_1 - \lambda_1} & \frac{\upsilon_1 - \omega_2}{\mu_1 - \lambda_2} & \cdots & \frac{\upsilon_1 - \omega_{n+1}}{\mu_1 - \lambda_{n+1}} \\ \frac{\upsilon_2 - \omega_1}{\mu_2 - \lambda_1} & \frac{\upsilon_2 - \omega_2}{\mu_2 - \lambda_2} & \cdots & \frac{\upsilon_2 - \omega_{n+1}}{\mu_2 - \lambda_{n+1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\upsilon_n - \omega_1}{\mu_n - \lambda_1} & \frac{\upsilon_n - \omega_2}{\mu_n - \lambda_2} & \cdots & \frac{\upsilon_n - \omega_{n+1}}{\mu_n - \lambda_{n+1}} \end{bmatrix} \in \mathbb{C}^{n \times (n+1)}, \tag{3.6}$$

and the coefficients a_i can be computed from the null space, e.g., the kernel of Loewner $ker(\mathbb{L}) = \mathbf{a}$. Having the coefficients a_i, we compute the b_i, and the rational function has been determined uniquely.

3.2.2. The Loewner pencil

Given a row array of complex numbers $(\mu_j, \mathbf{v}_j)$, $j = 1, \cdots, q$, and a column array, $(\lambda_i, \mathbf{w}_i)$, $i = 1, \cdots, k$, (with λ_i and the μ_j mutually distinct) the associated Loewner and shifted Loewner matrices are:

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{v}_1 - \mathbf{w}_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mathbf{v}_1 - \mathbf{w}_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_q - \mathbf{w}_1}{\mu_q - \lambda_1} & \cdots & \frac{\mathbf{v}_q - \mathbf{w}_k}{\mu_q - \lambda_k} \end{bmatrix} \in \mathbb{C}^{q \times k}, \ \mathbb{L}_s = \begin{bmatrix} \frac{\mu_1 \mathbf{v}_1 - \lambda_1 \mathbf{w}_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mu_1 \mathbf{v}_1 - \lambda_k \mathbf{w}_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mu_q \mathbf{v}_q - \lambda_1 \mathbf{w}_1}{\mu_q - \lambda_1} & \cdots & \frac{\mu_q \mathbf{v}_q - \lambda_k \mathbf{w}_k}{\mu_q - \lambda_k} \end{bmatrix} \in \mathbb{C}^{q \times k}.$$

Definition 3.1 (McMillan degree):
If $\mathbf{g}$ is rational, i.e., $\mathbf{g}(s) = \frac{\mathbf{p}(s)}{\mathbf{q}(s)}$, for appropriate polynomials $\mathbf{p}$, $\mathbf{q}$, the McMillan degree or the complexity of $\mathbf{g}$ is $\deg \mathbf{g} = \max\{\deg(\mathbf{p}), \deg(\mathbf{q})\}$. ◊

Now, if $\mathbf{w}_i = \mathbf{g}(\lambda_i)$, and $\mathbf{v}_j = \mathbf{g}(\mu_j)$, are *samples* of a rational function $\mathbf{g}$, the *main property* of Loewner matrices asserts the following.

Theorem 3.2 (Minimality of LTI systems):
[122] Let $\mathbb{L}$ be as above. If $k, q \geq deg\, \mathbf{g}$, then $rank\, \mathbb{L} = \deg \mathbf{g}$. In other words, the rank of $\mathbb{L}$ encodes the complexity of the underlying rational function $\mathbf{g}$. Furthermore, the same result holds for matrix-valued functions $\mathbf{g}$. ◊

3.2.3. Interpolatory projectors

We denote the tangential versions of (2.18) and (2.20) by $\mathcal{R}$, $\mathcal{O}$, respectively. For arbitrary k and q, these are defined as:

$$\mathcal{R} = \left[(\lambda_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_1, \ \cdots \ , \ (\lambda_k \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_k\right] \in \mathbb{C}^{n \times k}, \tag{3.7}$$

$$\mathcal{O}^T = \left[(\mu_1 \mathbf{E}^T - \mathbf{A}^T)^{-1} \mathbf{C}^T \ell_1 \ \ \cdots \ \ (\mu_q \mathbf{E}^T - \mathbf{A}^T)^{-1} \mathbf{C}^T \ell_q\right] \in \mathbb{C}^{n \times k}. \tag{3.8}$$

It readily follows that the reduced quantities $\hat{\mathbf{E}}$ and $\hat{\mathbf{A}}$ form a Loewner pencil:

$$\hat{\mathbf{E}} = -\mathcal{O} \mathbf{E} \mathcal{R} = -\begin{bmatrix} \frac{\mathbf{v}_1^T \mathbf{r}_1 - \ell_1^T \mathbf{w}_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mathbf{v}_1^T \mathbf{r}_k - \ell_1^T \mathbf{w}_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_q^T \mathbf{r}_1 - \ell_q^T \mathbf{w}_1}{\mu_q - \lambda_1} & \cdots & \frac{\mathbf{v}_q^T \mathbf{r}_k - \ell_q^T \mathbf{w}_k}{\mu_q - \lambda_k} \end{bmatrix} = -\mathbb{L} \in \mathbb{C}^{q \times k}, \tag{3.9}$$

$$\hat{\mathbf{A}} = -\mathcal{O} \mathbf{A} \mathcal{R} = -\begin{bmatrix} \frac{\mu_1 \mathbf{v}_1^T \mathbf{r}_1 - \ell_1^T \mathbf{w}_1 \lambda_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mu_1 \mathbf{v}_1^T \mathbf{r}_k - \ell_1^T \mathbf{w}_k \lambda_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mu_q \mathbf{v}_q^T \mathbf{r}_1 - \ell_q^T \mathbf{w}_1 \lambda_1}{\mu_q - \lambda_1} & \cdots & \frac{\mu_q \mathbf{v}_q^T \mathbf{r}_k - \ell_q^T \mathbf{w}_k \lambda_k}{\mu_q - \lambda_k} \end{bmatrix} = -\mathbb{L}_s \in \mathbb{C}^{q \times k}, \tag{3.10}$$

$$\hat{\mathbf{B}} = \mathcal{O}\mathbf{B} = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_q^T \end{bmatrix} = \mathbb{V} \in \mathbb{C}^{q\times m}\ , \quad \hat{\mathbf{C}} = \mathbf{C}\mathcal{R} = \left[\ \mathbf{w}_1 \quad \cdots \quad \mathbf{w}_k\ \right] = \mathbb{W} \in \mathbb{C}^{p\times k}. \quad (3.11)$$

The resulting quadruple $(\mathbb{W},\, \mathbb{L},\, \mathbb{L}_s,\, \mathbb{V})$ is called the Loewner quadruple.

Lemma 3.3 (Sylvester equations and Loewner):
Upon multiplication of the first equation in (2.21) with $\mathcal{O}$ on the left and the second by $\mathcal{R}$ on the right, we obtain:

$$\mathbb{L}_s - \mathbb{L}\,\mathbf{\Lambda} = \mathbb{V}\,\mathbf{R} \qquad \textit{and} \qquad \mathbb{L}_s - \mathbf{M}\,\mathbb{L} = \mathbf{L}\,\mathbb{W}. \qquad (3.12)$$

By adding/subtracting appropriate multiples of these expressions, it follows that the Loewner quadruple satisfies the Sylvester equations

$$\mathbf{M}\mathbb{L} - \mathbb{L}\,\mathbf{\Lambda} = \mathbb{V}\mathbf{R} - \mathbf{L}\mathbb{W} \quad \textit{and} \quad \mathbf{M}\,\mathbb{L}_s - \mathbb{L}_s\mathbf{\Lambda} = \mathbf{M}\mathbb{V}\mathbf{R} - \mathbf{L}\mathbb{W}\mathbf{\Lambda}. \qquad (3.13)$$

◊

Theorem 3.4 (Tangential interpolation):
Assume that the pencil $(\mathbb{L}_s,\ \mathbb{L})$ is regular[1]. Then $\mathbf{H}(s) = \mathbb{W}(\mathbb{L}_s - s\mathbb{L})^{-1}\mathbb{V}$, satisfies the tangential interpolation condition (2.22). ◊

Proof. Multiplying the first Sylvester equation by s and subtracting it from the second one, we get

$$\mathbf{M}(\mathbb{L}_s - s\mathbb{L}) - (\mathbb{L}_s - s\mathbb{L})\mathbf{\Lambda} = (\mathbf{M} - s\mathbf{I})\mathbb{V}\mathbf{R} - \mathbf{L}\mathbb{W}(\mathbf{\Lambda} - s\mathbf{I}).$$

Multiplying this equation by $\mathbf{e}_i$ on the right and setting $s = \lambda_i$, we obtain

$$(\mathbf{M} - \lambda_i\mathbf{I})(\mathbb{L}_s - \lambda_i\mathbb{L})\mathbf{e}_i = (\mathbf{M} - \lambda_i\mathbf{I})\mathbb{V}\mathbf{r}_i \ \Rightarrow\ (\mathbb{L}_s - \lambda_i\mathbb{L})\mathbf{e}_i = \mathbb{V}\mathbf{r}_i$$
$$\Rightarrow\ \mathbb{W}\mathbf{e}_i = \mathbb{W}(\mathbb{L}_s - \lambda_i\mathbb{L})^{-1}\mathbb{V}\mathbf{r}_i.$$

Thus $\mathbf{w}_i = \mathbf{H}(\lambda_i)\mathbf{r}_i$. Next, we multiply the above equation by $\mathbf{e}_j^T$ on the left and set $s = \mu_j$:

$$\mathbf{e}_j^T(\mathbb{L}_s - \mu_j\mathbb{L})(\mathbf{\Lambda} - \mu_j\mathbf{I}) = \mathbf{e}_j^T\mathbf{L}\mathbb{W}(\mathbf{\Lambda} - \mu_j\mathbf{I}) \ \Rightarrow\ \mathbf{e}_j^T(\mathbb{L}_s - \mu_j\mathbb{L}) = \boldsymbol{\ell}_j\mathbb{W}$$
$$\Rightarrow\ \mathbf{e}_j^T\mathbb{V} = \boldsymbol{\ell}_j^T\mathbb{W}(\mathbb{L}_s - \mu_j\mathbb{L})^{-1}\mathbb{V}.$$

Thus $\mathbf{v}_j^T = \boldsymbol{\ell}_j^T\mathbf{H}(\mu_j)$. □

Remark 3.5 (Parametrization of all interpolants):
With $\mathbf{K} \in \mathbb{C}^{p\times m}$, the Sylvester equations can be rewritten as

$$\mathbf{M}\mathbb{L} - \mathbb{L}\mathbf{\Lambda} = (\mathbb{V} - \mathbf{L}\mathbf{K})\mathbf{R} - \mathbf{L}(\mathbb{W} - \mathbf{K}\mathbf{R}) \quad \text{and}$$
$$\mathbf{M}(\mathbb{L}_s + \mathbf{L}\mathbf{K}\mathbf{R}) - (\mathbb{L}_s + \mathbf{L}\mathbf{K}\mathbf{R})\mathbf{\Lambda} = \mathbf{M}(\mathbb{V} - \mathbf{L}\mathbf{K})\mathbf{R} - \mathbf{L}(\mathbb{W} - \mathbf{K}\mathbf{R})\mathbf{\Lambda}.$$

These equations imply that $\left(\bar{\mathbb{W}}, \mathbb{L}, \bar{\mathbb{L}}_s, \bar{\mathbb{V}}\right)$ is an interpolant for all $\mathbf{K} \in \mathbb{C}^{p\times m}$, where $\bar{\mathbb{L}}_s = \mathbb{L}_s + \mathbf{L}\mathbf{K}\mathbf{R}$, $\bar{\mathbb{V}} = \mathbb{V} - \mathbf{L}\mathbf{K}$ and $\bar{\mathbb{W}} = \mathbb{W} - \mathbf{K}\mathbf{R}$. ◊

[1] The pencil $(\mathbb{L}_s, \mathbb{L})$ is called regular if there is at least one value of $\lambda \in \mathbb{C}$ such that $\det(\mathbb{L}_s - \lambda\mathbb{L}) \neq 0$.

3.2.4. Construction of interpolants

If the pencil $(\mathbb{L}_s, \mathbb{L})$ is regular, then $\mathbf{E} = -\mathbb{L}$, $\mathbf{A} = -\mathbb{L}_s$, $\mathbf{B} = \mathbb{V}$, $\mathbf{C} = \mathbb{W}$, is a minimal interpolant of the data, i.e., $\mathbf{H}(s) = \mathbb{W}(\mathbb{L}_s - s\mathbb{L})^{-1}\mathbb{V}$, interpolates the data. Otherwise, as shown in [122], problem (2.22) has a solution provided that

$$\operatorname{rank}\,[s\,\mathbb{L} - \mathbb{L}_s] = \operatorname{rank}\,[\mathbb{L},\ \mathbb{L}_s] = \operatorname{rank}\begin{bmatrix}\mathbb{L}\\ \mathbb{L}_s\end{bmatrix} = r,$$

for all $s \in \{\lambda_j\} \cup \{\mu_i\}$. Consider, then, the short SVDs:

$$[\mathbb{L},\ \mathbb{L}_s] = \mathbf{Y}\widehat{\Sigma}_r\tilde{\mathbf{X}}^*, \quad \begin{bmatrix}\mathbb{L}\\ \mathbb{L}_s\end{bmatrix} = \tilde{\mathbf{Y}}\Sigma_r\mathbf{X}^*,$$

where $\widehat{\Sigma}_r,\ \Sigma_r \in \mathbb{R}^{r\times r}$, $\mathbf{Y} \in \mathbb{C}^{q\times r}$, $\mathbf{X} \in \mathbb{C}^{k\times r}$, $\tilde{\mathbf{Y}} \in \mathbb{C}^{2q\times r}$, $\tilde{\mathbf{X}} \in \mathbb{C}^{r\times 2k}$.

Remark 3.6 (Numerical rank):
r can be chosen as the numerical rank (as opposed to the exact rank) of the Loewner pencil. For issues related to the rank, we refer the reader to [7], page 50, for details. ◊

Theorem 3.7 (Compressed model):
The quadruple $(\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C})$ of size $r \times r$, $r \times r$, $r \times m$, $p \times r$, given by:

$$\mathbf{E} = -\mathbf{Y}^T\mathbb{L}\mathbf{X}, \quad \mathbf{A} = -\mathbf{Y}^T\mathbb{L}_s\mathbf{X}, \quad \mathbf{B} = \mathbf{Y}^T\mathbb{V}, \quad \mathbf{C} = \mathbb{W}\mathbf{X},$$

is a descriptor realization of an (approximate) interpolant of the data with McMillan degree $r = \operatorname{rank}\mathbb{L}$. ◊

Remark 3.8 (Descriptor realization and the D term):
(a) The Loewner approach constructs a descriptor representation $(\mathbb{W}, \mathbb{L}, \mathbb{L}_s, \mathbb{V})$, of an underlying dynamical system exclusively from the data, with no further manipulations involved (i.e., matrix factorizations or inversions). The pencil $(\mathbb{L}_s, \mathbb{L})$ is singular and needs to be projected to a regular pencil $(\mathbf{A}, \mathbf{E})$. However, as shown in the mass-spring-damper example in Eq. 3.16, inversion can be replaced by generalized inversion.

(b) As mentioned, in the Loewner framework, by construction, $\mathbf{D}$ terms are absorbed in the other matrices of the realization. Extracting the $\mathbf{D}$ term involves an eigenvalue decomposition of $(\mathbb{L}_s, \mathbb{L})$. ◊

3.2.5. Interpolation property of reduced systems

Given a Loewner quadruple and the projection matrices[2] $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{n\times k}$, let the reduced quantities be

$$\hat{\mathbb{L}} = \mathbf{X}^*\mathbb{L}\mathbf{Y},\ \hat{\mathbb{L}}_s = \mathbf{X}^*\mathbb{L}_s\mathbf{Y},\ \hat{\mathbb{V}} = \mathbf{X}^*\mathbb{V},\ \hat{\mathbb{W}} = \mathbb{W}\mathbf{Y}.$$

We also consider the projected $\mathbf{L}$ and $\mathbf{R}$ matrices, namely $\hat{\mathbf{L}} = \mathbf{X}^*\mathbf{L}$, $\hat{\mathbf{R}} = \mathbf{R}\mathbf{Y}$. The question arises whether these reduced quantities also satisfy interpolation conditions. The answer is affirmative, and to show this, we proceed as follows.

[2] We call $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{n\times k}$ *projection matrices* as they are used for defining the *projector*: $\mathbf{X}(\mathbf{Y}^*\mathbf{X})^{-1}\mathbf{Y}^*$.

The associated $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{M}}$ must satisfy the projected equations resulting from (3.12), i.e.

$$\hat{\mathbb{L}}_s - \hat{\mathbb{L}}\,\hat{\mathbf{\Lambda}} = \hat{\mathbb{V}}\hat{\mathbf{R}} \qquad \textit{and} \qquad \hat{\mathbb{L}}_s - \hat{\mathbf{M}}\,\hat{\mathbb{L}} = \hat{\mathbf{L}}\hat{\mathbb{W}}. \tag{3.14}$$

Notice that the projected Loewner pencil is not in Loewner form. To achieve this, we proceed as follows. We need to diagonalize $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{M}}$. For this purpose we compute the following two generalized eigenvalue decompositions:

$$[\mathbf{D}_{\hat{\mathbf{\Lambda}}}, \mathbf{T}_{\hat{\mathbf{\Lambda}}}] = \text{eig}\left(\hat{\mathbb{L}}_s - \hat{\mathbb{V}}\hat{\mathbf{R}}, \hat{\mathbb{L}}\right) \ \text{ and } \ [\mathbf{D}_{\hat{\mathbf{M}}}, \mathbf{T}_{\hat{\mathbf{M}}}] = \text{eig}\left(\hat{\mathbb{L}}_s - \hat{\mathbf{L}}\hat{\mathbb{W}}, \hat{\mathbb{L}}\right).$$

These decompositions imply:

$$\hat{\mathbf{\Lambda}} = \mathbf{T}_{\hat{\Lambda}}\mathbf{D}_{\hat{\Lambda}}\mathbf{T}_{\hat{\Lambda}}^{-1} \ \text{ and } \ \hat{\mathbf{M}} = \mathbf{T}_{\hat{\mathbf{M}}}\mathbf{D}_{\hat{\mathbf{M}}}\mathbf{T}_{\hat{\mathbf{M}}}^{-1}, \tag{3.15}$$

where for simplicity, it is assumed that the matrices $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{M}}$ are diagonalizable.

It follows that the (diagonal) entries of $\mathbf{D}_{\hat{\mathbf{\Lambda}}}$ and $\mathbf{D}_{\hat{\mathbf{M}}}$ are the right frequencies and the left frequencies of the reduced system, respectively. Furthermore, straightforward calculations imply that the remaining quantities are as follows:

$$\left\{\begin{array}{lcllcl} \bar{\mathbb{L}}_s & = & \mathbf{T}_{\hat{\mathbf{M}}}^{-1}\,\hat{\mathbb{L}}_s\,\mathbf{T}_{\hat{\mathbf{\Lambda}}}, & \bar{\mathbb{L}} & = & \mathbf{T}_{\hat{\mathbf{M}}}^{-1}\,\hat{\mathbb{L}}\,\mathbf{T}_{\hat{\mathbf{\Lambda}}}, \\ \bar{\mathbb{V}} & = & \mathbf{T}_{\hat{\mathbf{M}}}^{-1}\,\hat{\mathbb{V}}, & \bar{\mathbf{L}} & = & \mathbf{T}_{\hat{\mathbf{M}}}^{-1}\,\hat{\mathbf{L}}, \\ \bar{\mathbb{W}} & = & \hat{\mathbb{W}}\,\mathbf{T}_{\hat{\mathbf{\Lambda}}}, & \bar{\mathbf{R}} & = & \hat{\mathbf{R}}\,\mathbf{T}_{\hat{\mathbf{\Lambda}}}. \end{array}\right.$$

Conclusion The right/left data triples for the reduced system are $\left(\mathbf{D}_{\hat{\mathbf{\Lambda}}}, \bar{\mathbb{W}}, \bar{\mathbf{R}}\right)$, and $\left(\mathbf{D}_{\hat{\mathbf{M}}}, \bar{\mathbb{V}}, \bar{\mathbf{L}}\right)$, respectively, while the associated Loewner pencil is $(\bar{\mathbb{L}}_s, \bar{\mathbb{L}})$.

3.2.6. Real interpolants and reduced models

Most often, the data are collected from real systems. In these cases if (s_i, ϕ_i) $s_i, \phi_i \in \mathbb{C}$, is a measurement pair; for the interpolants/reduced models to be real, the complex conjugate pair $(\bar{s}_i, \bar{\phi}_i)$, should also be included. Thus the left/right frequencies, besides real quantities, contain complex ones appearing in complex conjugate pairs. For instance, in the SISO (single-input, single-output) case, let the real measurement frequencies be $\sigma_i \in \mathbb{R}$, and the complex ones $\hat{\sigma}_i + \mathrm{j}\cdot\hat{\omega}_i$ where j denotes the imaginary unit. We split them into two sets, the left, and the right ones, respectively, making sure that each set is closed under complex conjugation:

$$\begin{array}{l} \mathbf{M} = \{\sigma_i,\ i = 1, \cdots, r_1;\ \hat{\sigma}_i \pm \mathrm{j}\cdot\hat{\omega}_i,\ i = 1, \ldots, r_3\}, \\ \mathbf{\Lambda} = \{\sigma_i,\ i = r_1 + 1, \ldots, r_1 + r_2;\ \hat{\sigma}_i \pm \mathrm{j}\cdot\hat{\omega}_i,\ i = r_3 + 1, \ldots, r_3 + r_4\}. \end{array}$$

Thus the left set has r_1 real frequencies and r_3 complex frequencies together with their complex conjugates (total $r_1 + 2r_3$ numbers). Similarly, the numbers for the right set are r_2 and r_4, i.e., it consists of $r_2 + 2r_4$ numbers. The quantities $\mathbb{W}$ and $\mathbb{V}$ are assembled in accordance with $\mathbf{M}$ and $\mathbf{\Lambda}$. In addition, let us define the matrices:

$$\mathbf{J}_\mu = \text{blkdiag}[\mathbf{I}_{r_1}, \overbrace{\mathbf{J}, \cdots, \mathbf{J}}^{r_3 \,\text{terms}}] \in \mathbb{C}^{(r_1+2r_3)\times(r_1+2r_3)},$$
$$\mathbf{J}_\lambda = \text{blkdiag}[\mathbf{I}_{r_2}, \underbrace{\mathbf{J}, \cdots, \mathbf{J}}_{r_4 \,\text{terms}}] \in \mathbb{C}^{(r_2+2r_4)\times(r_2+2r_4)},$$

where $\mathbf{J} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -\mathrm{j} \\ 1 & \mathrm{j} \end{bmatrix}$, where blkdiag[$\cdot$] (following Matlab notation) denotes the diagonal block structure. A simple calculation shows then that the matrices.

$$\mathbf{M}_R = \mathbf{J}_\mu^* \mathbf{M} \mathbf{J}_\mu, \quad \mathbb{V}_R = \mathbf{J}_\mu^* \mathbb{V}, \quad \mathbf{L}_R = \mathbf{J}_\mu^* \mathbf{L},$$

Have real entries. The same happens with the matrices.

$$\mathbf{\Lambda}_R = \mathbf{J}_\lambda^* \mathbf{\Lambda} \mathbf{J}_\lambda, \quad \mathbb{W}_R = \mathbb{W} \mathbf{J}_\lambda, \quad \mathbf{R}_R = \mathbf{R} \mathbf{J}_\lambda.$$

Recall equations (3.12). If we now solve the transformed equations for $\mathbb{L}^R$, $\mathbb{L}_s^R$:

$$\mathbb{L}_s^R - \mathbb{L}^R \mathbf{\Lambda}_R = \mathbb{V}_R \mathbf{R}_R \quad \textit{and} \quad \mathbb{L}_s^R - \mathbf{M}_R \mathbb{L}^R = \mathbf{L}_R \mathbb{W}_R,$$

the resulting pencil ($\mathbb{L}_s^R$, $\mathbb{L}^R$) has real entries. Hence the algorithms based on $\mathbb{L}^R$ and $\mathbb{L}_s^R$ described below yield real reduced order models.

3.2.7. Algorithms for SISO and MIMO Loewner models

3.3. Introductory examples

In this section, the theory will be illustrated using simple examples.

Example 3.1 (A spring-mass-damper system):
Let m, d, and k denote the mass, damping, and stiffness of the spring; let also $x(t)$ denote the displacement and $F(t)$ the force applied; the associated differential equation is:

$$m\ddot{x}(t) + d\dot{x}(t) + kx(t) = F(t)$$

This is a SISO (single-input, single-output) system. By introducing the state variables

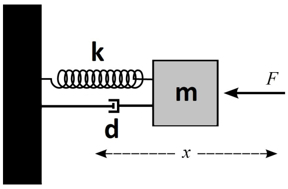


Figure 3.1.: A spring mass damper system.

Algorithm 3.1: The Loewner-SISO rational interpolation method

```
Input: $\mathbf{S} = [s_1, \cdots, s_N] \in \mathbb{C}^N,\ \mathbf{F} = [\phi_1, \cdots, \phi_N] \in \mathbb{C}^N,\ N \in \mathbb{N}$.
Output: $\hat{\mathbf{E}} \in \mathbb{R}^{r\times r}, \hat{\mathbf{A}} \in \mathbb{R}^{r\times r}, \hat{\mathbf{B}} \in \mathbb{R}^{r\times 1}, \hat{\mathbf{C}} \in \mathbb{R}^{1\times r}$ with $r \ll N$.
  Partition the measurements into 2 disjoint sets and form left and right set as:
  $(\mu_j, \mathbf{v}_j),\ j = 1, \ldots, q$ and $(\lambda_i, \mathbf{w}_i),\ i = 1, \ldots, k$.
      $\text{frequencies}: \quad [s_1, \cdots, s_N] \rightarrow [\lambda_1, \cdots, \lambda_k],\ [\mu_1, \cdots, \mu_q], \quad k + q = N,$
      $\text{values}: \quad [\phi_1, \cdots, \phi_N] \rightarrow [\mathbf{w}_1, \cdots, \mathbf{w}_k] = \mathbb{W},\ [\mathbf{v}_1, \cdots, \mathbf{v}_q] = \mathbb{V}^T.$
  Construct the Loewner pencil as:
      $\mathbb{L} = \left(\frac{\mathbf{v}_i - \mathbf{w}_j}{\mu_i - \lambda_j}\right)_{i=1,\ldots,q}^{j=1,\ldots,k}, \qquad \mathbb{L}_s = \left(\frac{\mu_i \mathbf{v}_i - \lambda_j \mathbf{w}_j}{\mu_i - \lambda_j}\right)_{i=1,\ldots,q}^{j=1,\ldots,k}.$
  It follows that the complex raw model is:
      $\{\mathbb{W}, \mathbb{L}, \mathbb{L}_s, \mathbb{V}\}.$
  Transform all the complex data to real, and it follows the raw, real model:
      $\{\mathbb{W}_R, \mathbb{L}^R, \mathbb{L}_s^R, \mathbb{V}_R\}.$
  Compute the rank revealing SVDs: $[\mathbf{Y}_1,\Sigma_1,\mathbf{X}_1] = \mathbf{SVD}([\mathbb{L}^R\ \mathbb{L}_s^R])$ and
  $[\mathbf{Y}_2,\Sigma_2,\mathbf{X}_2] = \mathbf{SVD}([\mathbb{L}^R;\mathbb{L}_s^R])$; the decay of the singular values, leads to the choice
  of the order $r$ of the approximant.
  The reduced real model is obtained by projecting the *raw real model* with
  $\mathbf{Y} = \mathbf{Y}_1^{n\times r}$ and $\mathbf{X} = \mathbf{X}_2^{n\times r}$ as:
      $\underbrace{\{\mathbb{W}_R, \mathbb{L}^R, \mathbb{L}_s^R, \mathbb{V}_R\}}_{\text{singular}} \underbrace{\Rightarrow}_{\text{SVD}} \underbrace{\{\mathbb{W}_R\mathbf{X}, \mathbf{Y}^T\mathbb{L}^R\mathbf{X}, \mathbf{Y}^T\mathbb{L}_s^R\mathbf{X}, \mathbf{Y}^T\mathbb{V}_R\}}_{\text{regular}} = \{\hat{\mathbf{C}}, -\hat{\mathbf{E}}, -\hat{\mathbf{A}}, \hat{\mathbf{B}}\}.$
  A real approximant of the data is then:
      $\hat{H}(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} \approx \phi(s).$
```

Algorithm 3.2: The Loewner MIMO rational interpolation method

```
Input: $\mathbf{S} = [s_1, \cdots, s_N] \in \mathbb{C}^N, \mathbf{F} = [\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_N] \in \mathbb{C}^{N\times p\times m}, N \in \mathbb{N}$.
Output: $\hat{\mathbf{E}} \in \mathbb{R}^{r\times r}, \hat{\mathbf{A}} \in \mathbb{R}^{r\times r}, \hat{\mathbf{B}} \in \mathbb{R}^{r\times m}, \hat{\mathbf{C}} \in \mathbb{R}^{p\times r}$ with $r \ll N$.
1: Partition the measurements into 2 disjoint sets:
   Left data:
   $$\mathbf{M} = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_q \end{bmatrix} \in\mathbb{C}^{q\times q}, \quad \mathbf{L} = \begin{bmatrix} \boldsymbol{\ell}_1^T \\ \vdots \\ \boldsymbol{\ell}_q^T \end{bmatrix} \in\mathbb{C}^{q\times p}, \quad \mathbb{V} = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_q^T \end{bmatrix} \in \mathbb{C}^{q\times m}.$$
   Right data:
   $$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_k \end{bmatrix} \in \mathbb{C}^{k\times k}, \qquad \begin{aligned} \mathbf{R} &= [\mathbf{r}_1, \;\; \mathbf{r}_2, \;\; \cdots \;\; \mathbf{r}_k] \in \mathbb{C}^{m\times k}, \\ \mathbb{W} &= [\mathbf{w}_1 \;\; \mathbf{w}_2 \;\; \cdots \;\; \mathbf{w}_k] \in \mathbb{C}^{p\times k}. \end{aligned}$$
2: Construct the Loewner pencil as:
   $$\mathbb{L} = \left( \frac{\mathbf{v}_i^T \mathbf{r}_i - \ell_j^T \mathbf{w}_j}{\mu_i - \lambda_j} \right)_{i=1,\ldots,q}^{j=1,\ldots,k}, \qquad \mathbb{L}_s = \left( \frac{\mu_i \mathbf{v}_i^T \mathbf{r}_i - \lambda_j \ell_j^T \mathbf{w}_j}{\mu_i - \lambda_j} \right)_{i=1,\ldots,q}^{j=1,\ldots,k}.$$
3: It follows that the complex raw model is:
   $$\{\mathbb{W}, \mathbb{L}, \mathbb{L}_s, \mathbb{V}\}.$$
3: Transform all the complex data to real, and it follows the real raw model:
   $$\{\mathbb{W}_R, \mathbb{L}^R, \mathbb{L}_s^R, \mathbb{V}_R\}.$$
4: Compute the rank revealing SVDs: $[\mathbf{Y}_1,\Sigma_1,\mathbf{X}_1] = \mathbf{SVD}([\mathbb{L}^R \; \mathbb{L}_s^R])$ and
   $[\mathbf{Y}_2,\Sigma_2,\mathbf{X}_2] = \mathbf{SVD}([\mathbb{L}^R; \mathbb{L}_s^R])$; the decay of the singular values, lead to the choice of $r$.
5: The reduced real model is obtained by projecting the *raw real model* with:
   $\mathbf{Y} = \mathbf{Y}_1^{n\times r}$ and $\mathbf{X} = \mathbf{X}_2^{n\times r}$.
   $$\underbrace{\{\mathbb{W}_R, \mathbb{L}^R, \mathbb{L}_s^R, \mathbb{V}_R\}}_{\text{singular}} \underbrace{\Rightarrow}_{\text{SVD}} \underbrace{\{\mathbb{W}_R\mathbf{X}, \mathbf{Y}^T\mathbb{L}^R\mathbf{X}, \mathbf{Y}^T\mathbb{L}_s^R\mathbf{X}, \mathbf{Y}^T\mathbb{V}_R\}}_{\text{regular}} = \{\hat{\mathbf{C}}, -\hat{\mathbf{E}}, -\hat{\mathbf{A}}, \hat{\mathbf{B}}\}.$$
6: A real approximant of the data is:
   $$\hat{\mathbf{H}}(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} \approx \boldsymbol{\phi}(s).$$
```

$x_1 = x,\ x_2 = \dot{x}$, the input $u = F$, and as output the velocity $y = \dot{x}$, the following state equations result:

$$\dot{x}_1(t) = x_2(t), \quad m\dot{x}_2(t) = -kx_1 - dx_2(t) + u(t), \quad y(t) = x_2(t).$$

The system matrices are thus:

$$\mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & m \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k & -d \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

and the resulting transfer function:

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} = \frac{s}{ms^2 + ds + k}.$$

Next, we will assume for simplicity that all parameters have value one. We now wish to recover state equations equivalent to the above from transfer function measurements. Toward this goal, we evaluate the transfer function at the real frequencies: $\lambda_1 = \frac{1}{2}$, $\lambda_2 = 1$ (right frequencies), as well as $\mu_1 = -\frac{1}{2}$, $\mu_2 = -1$ (left frequencies). The corresponding values of $\mathbf{H}$ are collected in the matrices.

$$\mathbb{W} = \left(\tfrac{2}{7} \quad \tfrac{1}{3} \right), \quad \mathbb{V} = \left(-\tfrac{2}{3} \quad -1 \right)^T.$$

Furthermore with $\mathbf{R} = [1 \ \ 1] = \mathbf{L}^T$, we construct the Loewner pencil:

$$\mathbb{L} = \begin{bmatrix} \frac{20}{21} & \frac{2}{3} \\ \frac{6}{7} & \frac{2}{3} \end{bmatrix}, \quad \mathbb{L}_s = \begin{bmatrix} -\frac{4}{21} & 0 \\ -\frac{4}{7} & -\frac{1}{3} \end{bmatrix}.$$

Since the pencil $(\mathbb{L}_s,\ \mathbb{L})$ is regular, we recover the original transfer function:

$$\mathbf{H}(s) = \mathbb{W}\mathbf{\Phi}(s)^{-1}\mathbb{V} = \frac{s}{s^2 + s + 1}, \quad \text{where} \quad \mathbf{\Phi}(s) = \mathbb{L}_s - s\,\mathbb{L}.$$

Hence, the measurements above yield a minimal (descriptor) realization of the system in terms of the (state) variables $\mathbf{z}_1$, $\mathbf{z}_2$:

$$\begin{aligned} \tfrac{20}{21}\,\dot{\mathbf{z}}_1(t) + \tfrac{2}{3}\,\dot{\mathbf{z}}_2(t) &= -\tfrac{4}{21}\mathbf{z}_1(t) + \tfrac{2}{3}\mathbf{u}(t), \\ \tfrac{6}{7}\,\dot{\mathbf{z}}_1(t) + \tfrac{2}{3}\,\dot{\mathbf{z}}_2(t) &= -\tfrac{4}{7}\,\mathbf{z}_1(t) - \tfrac{1}{3}\,\mathbf{z}_2(t) + \mathbf{u}(t), \\ \mathbf{y}(t) &= \tfrac{2}{7}\,\mathbf{z}_1(t) + \tfrac{1}{3}\,\mathbf{z}_2(t). \end{aligned}$$

with

$$\tilde{\mathbf{E}} = \begin{bmatrix} \frac{20}{21} & \frac{2}{3} \\ \frac{6}{7} & \frac{2}{3} \end{bmatrix}, \quad \tilde{\mathbf{A}} = \begin{bmatrix} -\frac{4}{21} & 0 \\ -\frac{4}{7} & -\frac{1}{3} \end{bmatrix}, \quad \tilde{\mathbf{B}} = \begin{bmatrix} \frac{2}{3} \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{C}} = \begin{bmatrix} \frac{2}{7} & \frac{1}{3} \end{bmatrix}.$$

By multiplying with $\tilde{\mathbf{E}}^{-1}$, it yields: (id: identified system in state space form)

$$\tilde{\mathbf{A}}_{id} = \begin{bmatrix} 4 & \frac{7}{2} \\ -6 & -5 \end{bmatrix}, \quad \tilde{\mathbf{B}}_{id} = \begin{bmatrix} -\frac{7}{2} \\ 6 \end{bmatrix}, \quad \tilde{\mathbf{C}}_{id} = \begin{bmatrix} \frac{2}{7} & \frac{1}{3} \end{bmatrix}.$$

Coordinate tranformation Let the state vector $\mathbf{x}$ be transformed to the new state vector $\mathbf{z}$ by the non-singular transformation matrix

$$\mathbf{\Psi} = \left[\begin{array}{c} \mathbf{C} \\ \mathbf{CA} \end{array}\right]^{-1} \left[\begin{array}{c} \tilde{\mathbf{C}}_{id} \\ \tilde{\mathbf{C}}_{id}\,\tilde{\mathbf{A}}_{id} \end{array}\right],$$

of dimension 2×2. Then, the following hold:

$$\mathbf{z} = \mathbf{\Psi}^{-1}\mathbf{x},\ \tilde{\mathbf{A}}_{id} = \mathbf{\Psi}^{-1}\mathbf{A}\mathbf{\Psi},\ \tilde{\mathbf{B}}_{id} = \mathbf{\Psi}^{-1}B,\ \tilde{\mathbf{C}}_{id} = \mathbf{C}\mathbf{\Psi}.$$

(e.g., $\mathbf{\Psi}\tilde{A}_{id}\mathbf{\Psi}^{-1} = \left[\begin{array}{cc} 0 & 1 \\ -1 & -1 \end{array}\right] = \mathbf{A}$).

Remark 3.9 (Invariant information and identification):
The above result certifies that the Loewner framework constitutes a data-driven system identification method that constructs a realization only from measurements. It is essential to mention that initial and identified systems are identical under a coordinate transformation. At the same time, the underlying dynamics are recovered exactly while the corresponding revealing transfer function remains invariant under such a transformation. ◊

The question now arises: what happens if we collect more data than necessary:

$$\mathbf{\Lambda} = \text{diag}\left(\begin{array}{cccc} \frac{1}{2} & 1 & \frac{3}{2} & 2 \end{array}\right),\quad \mathbf{M} = \text{diag}\left(\begin{array}{cccc} -\frac{1}{2} & -1 & -\frac{3}{2} & -2 \end{array}\right).$$

In this case, the associated measurements are

$$\mathbb{W} = \left(\begin{array}{cccc} \frac{2}{7} & \frac{1}{3} & \frac{6}{19} & \frac{2}{7} \end{array}\right),\quad \mathbb{V} = \left(\begin{array}{cccc} -\frac{2}{3} & -1 & -\frac{6}{7} & -\frac{2}{3} \end{array}\right)^T,$$

and with $\mathbf{R} = [1\ \ 1\ \ 1\ \ 1] = \mathbf{L}^T$, the Loewner pencil is:

$$\mathbb{L} = \left[\begin{array}{cc|cc} \frac{20}{21} & \frac{2}{3} & \frac{28}{57} & \frac{8}{21} \\ \frac{6}{7} & \frac{2}{3} & \frac{10}{19} & \frac{3}{7} \\ \hline \frac{4}{7} & \frac{10}{21} & \frac{52}{133} & \frac{16}{49} \\ \frac{8}{21} & \frac{1}{3} & \frac{16}{57} & \frac{5}{21} \end{array}\right],\quad \mathbb{L}_s = \left[\begin{array}{cc|cc} -\frac{4}{21} & 0 & \frac{4}{57} & \frac{2}{21} \\ -\frac{4}{7} & -\frac{1}{3} & -\frac{4}{19} & -\frac{1}{7} \\ \hline -\frac{4}{7} & -\frac{8}{21} & -\frac{36}{133} & -\frac{10}{49} \\ -\frac{10}{21} & -\frac{1}{3} & -\frac{14}{57} & -\frac{4}{21} \end{array}\right].$$

It turns out that we can choose arbitrary matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{4\times 2}$, provided that $\det\left(\mathbf{Y}^T\mathbf{X}\right) \neq 0$, e.g.

$$\mathbf{X} = \left[\begin{array}{cc} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ -2 & 1 \end{array}\right],\quad \mathbf{Y}^T = \left[\begin{array}{cccc} 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right],$$

so that the projected quantities

$$\widehat{\mathbb{W}} = \mathbb{W}\mathbf{X} = \left[\begin{array}{cc} -\frac{6}{7} & -\frac{1}{21} \end{array}\right],\ \widehat{\mathbb{L}} = \mathbf{Y}^T\mathbb{L}\mathbf{X} = \left[\begin{array}{cc} -\frac{6}{7} & -\frac{1}{7} \\ \frac{18}{49} & \frac{1}{147} \end{array}\right],$$

$$\widehat{\mathbb{L}}_s = \mathbf{Y}^T \mathbb{L}_s \mathbf{X} = \begin{bmatrix} 0 & \frac{1}{21} \\ -\frac{48}{49} & -\frac{19}{147} \end{bmatrix}, \quad \widehat{\mathbb{V}} = \mathbf{Y}^T \mathbb{V} = \begin{bmatrix} -\frac{1}{3} \\ \frac{11}{21} \end{bmatrix},$$

constitute a minimal realization of $\mathbf{H}(s)$:

$$\mathbf{H}(s) = \widehat{\mathbb{W}} \left(\widehat{\mathbb{L}}_s - s\widehat{\mathbb{L}} \right)^{-1} \widehat{\mathbb{V}} \; = \frac{s}{s^2+s+1}.$$

It should be stressed that this holds for arbitrary projection matrices $\mathbf{X}$, $\mathbf{Y}$. The quantity needed is the generalized inverse Section 2.1.3 of

$$\mathbf{\Phi}(s) = \mathbb{L}_s - s\,\mathbb{L} = \begin{bmatrix} -\frac{20s}{21} - \frac{4}{21} & -\frac{2s}{3} & \frac{4}{57} - \frac{28s}{57} & \frac{2}{21} - \frac{8s}{21} \\ -\frac{6s}{7} - \frac{4}{7} & -\frac{2s}{3} - \frac{1}{3} & -\frac{10s}{19} - \frac{4}{19} & -\frac{3s}{7} - \frac{1}{7} \\ -\frac{4s}{7} - \frac{4}{7} & -\frac{10s}{21} - \frac{8}{21} & -\frac{52s}{133} - \frac{36}{133} & -\frac{16s}{49} - \frac{10}{49} \\ -\frac{8s}{21} - \frac{10}{21} & -\frac{s}{3} - \frac{1}{3} & -\frac{16s}{57} - \frac{14}{57} & -\frac{5s}{21} - \frac{4}{21} \end{bmatrix}. \tag{3.16}$$

We first notice that $\mathbf{\Phi}(s) = \mathbf{X}\mathbf{\Delta}(s)\mathbf{Y}^T$, where $\mathbf{X}$ and $\mathbf{Y}$ can be chosen as follows

$$\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{3}{7} & \frac{8}{7} \\ -\frac{1}{2} & 1 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 1 & 0 & -\frac{7}{19} & -\frac{1}{2} \\ 0 & 1 & \frac{24}{19} & \frac{9}{7} \end{bmatrix} \quad \Rightarrow \quad \det\left(\mathbf{YX}\right) \neq 0.$$

Thus by taking the 2×2 upper-left block as $\mathbf{\Delta}(s) = \mathbf{\Phi}(1:2,1:2)(s)$, it follows that $\mathbf{\Phi}(s)^{MP} = \frac{1}{80989667}\frac{1}{s^2+s+1}\mathbf{Z}(s)$, where $\mathbf{Z}(s) =$

$$\begin{bmatrix} -28\,(11610185s + 7274073) & 14\,(3558666s - 5604037) & 6076\,(32301s - 391) & 14\,(15168851s + 1670036) \\ 294\,(225182s + 281171) & (-147)\,(192415s - 19668) & -2058\,(29494s + 15609) & -147\,(417597s + 261503) \\ 3724\,(54617s + 48189) & (-1862)\,(29046s - 17485) & -26068\,(5715s + 1523) & -1862\,(83663s + 30704) \\ 98\,(2527157s + 2123670) & -49\,(1250553s - 876439) & -98\,(1797669s + 409322) & -49\,(3777710s + 1247231) \end{bmatrix}.$$

In the rectangular case, where there are two fewer right measurements, i.e., we only have $\widetilde{\mathbf{\Lambda}} = \text{diag}\left[\frac{1}{2},\;\; 1\right]$, while $\mathbf{M}$ remains the same, the right values are $\widetilde{\mathbb{W}} = \mathbb{W}(:,1\;:\;2)$; hence

$$\widetilde{\mathbf{\Phi}}(s) = \widetilde{\mathbb{L}}_s - s\,\widetilde{\mathbb{L}} = \left[\begin{array}{c|c} -\frac{20s}{21} - \frac{4}{21} & -\frac{2s}{3} \\ -\frac{6s}{7} - \frac{4}{7} & -\frac{2s}{3} - \frac{1}{3} \\ -\frac{4s}{7} - \frac{4}{7} & -\frac{10s}{21} - \frac{8}{21} \\ -\frac{8s}{21} - \frac{10}{21} & -\frac{s}{3} - \frac{1}{3} \end{array}\right] = \mathbf{X}\,\mathbf{\Delta}(s)\,\widetilde{\mathbf{Y}}^T,$$

has dimension 4×2, where $\widetilde{\mathbf{Y}} = \mathbf{Y}(1:2,1:2)$. In this case, the Moore-Penrose Section 2.1.3 inverse is

$$\widetilde{\mathbf{\Phi}}(s)^{MP} = \frac{1}{737\,(s^2+s+1)} \left[\begin{array}{cccc} -4767s - 3402 & \frac{1827}{2}s - \frac{2037}{2} & 3087s + 294 & 3297s + \frac{1365}{2} \\ \hline 5838s + 5250 & -1596s + 903 & -4326s - 1218 & -4515s - 1722 \end{array}\right],$$

which implies the desired equality

$$\Rightarrow \mathbb{W}\,\mathbf{\Phi}(s)^{MP}\,\mathbb{V} \;=\; \widetilde{\mathbb{W}}\,\widetilde{\mathbf{\Phi}}(s)^{MP}\,\mathbb{V} \;=\; \mathbf{H}(s).$$

Conclusion: the Loewner framework allows the definition of rectangular and singular systems. ◊

Example 3.2 (Reduction of a 10th order band-stop filter):
The system has two inputs and outputs (MIMO), state-space dimension 10, and a $\mathbf{D}$ term of rank 2. A state-space representation is as follows:

$$\boldsymbol{\Sigma}: \quad \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t), \qquad \text{where}$$

$$\mathbf{A} = \left[\begin{array}{ccccc|ccccc} -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & -1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & -1 & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & -1 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & -1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \ \mathbf{B} = \left[\begin{array}{cc} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right],$$

$$\mathbf{C} = \left[\begin{array}{ccccc|ccccc} -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \end{array}\right], \ \mathbf{D} = \left[\begin{array}{cc} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{array}\right].$$

The transfer function is a 2×2 rational matrix given by:

$$\mathbf{H}(s) = \frac{1}{\mathbf{d}(s)} \left[\begin{array}{cc} \mathbf{n}_1(s) & \mathbf{n}_2(s) \\ -\mathbf{n}_2(s) & -\mathbf{n}_1(s) \end{array}\right] + \mathbf{D}, \qquad \text{where}$$

$$\begin{aligned} \mathbf{n}_1(s) &= s\left(s^8 + 7s^6 + 13s^4 + 7s^2 + 1\right), \\ \mathbf{n}_2(s) &= s\left(5s^8 + 6s^7 + 25s^6 + 20s^5 + 41s^4 + 20s^3 + 25s^2 + 6s + 5\right), \\ \mathbf{d}(s) &= 2(s^4 + s^3 + 3s^2 + 2s+)(2s^6 + 3s^5 + 7s^4 + 7s^3 + 7s^2 + 3s + 2). \end{aligned}$$

It readily follows that $\lim_{s\to\infty} \mathbf{H}(s) = \mathbf{D}$. We take $N = 100$ samples of the transfer function on the imaginary axis (frequency response measurements) between 10^{-1} and 10^{1} rad/sec. Fig. 3.3 (left) shows the first 20 normalized singular values of the resulting real Loewner pencil (the rest are numerically zero). The rank of $\mathbb{L}$ is 10 (the McMillan degree of the system), while the rank of $\mathbb{L}_s$ is 12 ($= \operatorname{rank}\mathbb{L} + \operatorname{rank}\mathbf{D}$). The right pane in Fig. 3.2 shows that we can obtain a perfect fit (total recovery of the model) with the Loewner framework for this MIMO example only by sampling the transfer function. As both grammians are: $\mathcal{P} = \mathcal{Q} = \frac{1}{2}\mathbf{I}_{10}$, i.e., equal and a multiple of the identity matrix, the Hankel singular values (see [7]) are all equal; this makes reduction with balanced truncation not feasible.

The right pane in Fig. 3.3 shows the poles of the system obtained using the Loewner framework, along with the zeros for every entry. The right pane in Fig. 3.2 shows the band-stop character around frequency $\omega_0 = 1\ rad/s$, of entries $(1,2)$ and $(2,1)$.

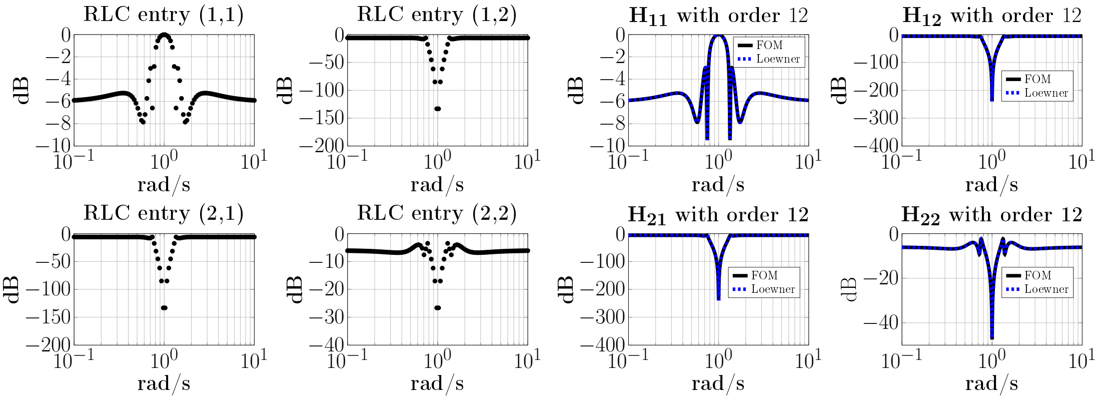


Figure 3.2.: Left: Shows the 100 measurements sampled with DNS(Direct Numerical Simulations) of the theoretical (2×2)-matrix transfer function. Right: Loewner approximants.

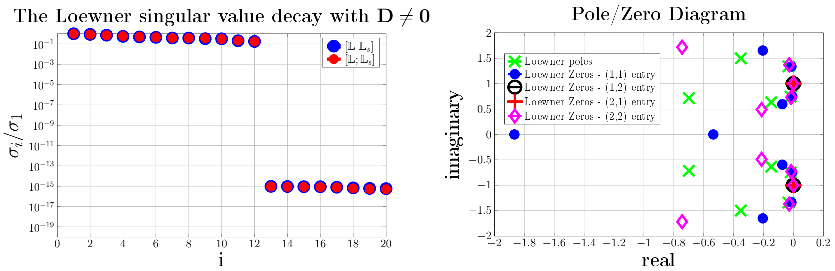


Figure 3.3.: Left: Shows the first 12 singular values while the rest are numerically zero. Right: Pole/Zero diagram.

Computing the poles of the Loewner model confirms the accuracy of the approach.

eig $(\mathbf{A})$	eig $(\mathbf{A}_r, \mathbf{E}_r)$
$-0.0181885913675508 - 0.745231200229\,\mathrm{i}$	$-0.0181885913675508 - 0.745231200229\,\mathrm{i}$
$-0.0181885913675508 + 0.745231200229\,\mathrm{i}$	$-0.0181885913675508 + 0.745231200229\,\mathrm{i}$
$-0.148402943598342 - 0.632502179219046\,\mathrm{i}$	$-0.148402943598342 - 0.632502179219046\,\mathrm{i}$
$-0.148402943598342 + 0.632502179219046\,\mathrm{i}$	$-0.148402943598342 + 0.632502179219046\,\mathrm{i}$
$-0.699080475814867 - 0.715042997542469\,\mathrm{i}$	$-0.699080475814867 - 0.715042997542469\,\mathrm{i}$
$-0.699080475814867 + 0.715042997542469\,\mathrm{i}$	$-0.699080475814867 + 0.715042997542469\,\mathrm{i}$
$-0.0327309328175858 - 1.34106659803138\,\mathrm{i}$	$-0.0327309328175858 - 1.34106659803138\,\mathrm{i}$
$-0.0327309328175858 + 1.34106659803138\,\mathrm{i}$	$-0.0327309328175858 + 1.34106659803138\,\mathrm{i}$
$-0.351597056401658 - 1.49852758300335\,\mathrm{i}$	$-0.351597056401658 - 1.49852758300335\,\mathrm{i}$
$-0.351597056401658 + 1.49852758300335\,\mathrm{i}$	$-0.351597056401658 + 1.49852758300335\,\mathrm{i}$
	∞
	∞

As can be observed from this table, the Loewner method computes, besides the finite poles, two poles at infinity. This happens because, in the Loewner framework, the bD term is incorporated in the remaining matrices of the realization. ◊

3.4. Summary of the method

The following result summarizes the cases that arise in the Loewner framework, depending on the amount of data available.

Lemma 3.10:
Given is a scalar transfer function of McMillan degree n.

1. Amount of data less that $2n$. For $q = k \leq n$, define the transfer function $\hat{\mathbf{H}}(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}}$, using the Loewner procedure. The interpolation conditions below are satisfied:

$$\hat{\mathbf{H}}(\mu_i) = \mathbf{H}(\mu_i) \ \text{ and } \ \hat{\mathbf{H}}(\lambda_i) = \mathbf{H}(\lambda_i) \ \text{ for } \ i = 1, \ldots, k.$$

   If $k = q = n$, the Loewner quadruple is equivalent to the original one $(\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$.

2. Arbitrary amount of data, no reduction. For arbitrary k and q (i.e. k, $q \leq n$ or k, $q \geq n$) the Loewner quadruple interpolates the data, even if the pencil $(\mathbb{L}_s, \mathbb{L})$ is singular. This is to be interpreted as follows:

$$(\mathbb{L}_s - \lambda_i \mathbb{L})\, \mathbf{e}_i = \mathbb{V} \quad \textit{and} \quad \mathbf{e}_j^T\, (\mathbb{L}_s - \mu_j \mathbb{L}) = \mathbb{W}.$$

   Hence $\mathbb{W}\mathbf{e}_i = \mathbf{w}_i$, $i = 1, \cdots, k$, and $\mathbf{e}_j^T \mathbb{V} = \mathbf{v}_j$, $j = 1, \cdots, q$. Therefore the transfer function of the Loewner pencil interpolates $\mathbf{H}(s)$ at the left and right interpolation points.

3. Arbitrary amount of data, followed by reduction. If k, $q \geq n$, consider the rank revealing SVD decompositions:

$$[\mathbb{L} \quad \mathbb{L}_s] = \hat{\mathbf{Y}}_r \hat{\Sigma}_r \mathbf{X}_r^T \ \text{ and } \ \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \mathbf{Y}_r \tilde{\Sigma}_r \tilde{\mathbf{X}}_r^T,$$

   where $\mathbf{Y}_r \in \mathbb{R}^{q \times r}$, $\mathbf{X}_r \in \mathbb{R}^{k \times r}$, and $r \leq k$, q, is the exact or the numerical rank of the Loewner pencil involved. Let

$$\tilde{\mathbf{E}} = \mathbf{Y}_r^T \mathbb{L} \mathbf{X}_r,\ \tilde{\mathbf{A}} = \mathbf{Y}_r^T \mathbb{L}_s \mathbf{X}_r \ \in \ \mathbb{C}^{r \times r},\ \tilde{\mathbf{B}} = \mathbf{Y}_r^T \mathbb{V} \ \in \ \mathbb{C}^r,\ \tilde{\mathbf{C}} = \mathbb{W}\mathbf{X}_r \ \in \ \mathbb{C}^{1 \times r}.$$

   Then the following approximate interpolation conditions are satisfied:

$$\tilde{\mathbf{H}}(\mu_i) \approx \mathbf{H}(\mu_i),\ i = 1, \cdots, q, \quad \text{and} \quad \tilde{\mathbf{H}}(\lambda_j) \approx \mathbf{H}(\lambda_j),\ j = 1, \cdots, k.$$

   In addition, the reduced system satisfies (exact) interpolation conditions as shown in section 3.2.5. ◊

3.5. Practical considerations and some advances

In this section, we apply the CUR factorization to the Loewner matrix. We follow [113], where CUR is applied to Hankel matrices instead.

Definition 3.11:
With $\mathbb{L} \in \mathbb{R}^{n\times n}$, let $\mathcal{I} = \{i_1, \ldots, i_r\}$ and $\mathcal{J} = \{j_1, \ldots, j_r\}$ denote the r-subsets ($r \ll n$) of row and column indices respectively. If $(\cdot)^{MP}$ denotes the pseudo inverse, then the CUR factorization of the Loewner matrix $\mathbb{L}$ is given by

$$\mathbb{L}_r := \underbrace{\mathbb{L}(:,\mathcal{J})}_{\mathcal{J}\text{-columns}} \cdot \mathbb{L}(\mathcal{I},\mathcal{J})^{\mathrm{MP}} \cdot \underbrace{\mathbb{L}(\mathcal{I},:)}_{\mathcal{I}\text{-rows}}. \tag{3.17}$$

◊

In practical applications, large-scale data matrices are only approximately low-rank (when data can be, for instance, corrupted by noise). In this case, the sets $\mathcal{I}$ and $\mathcal{J}$ need to be chosen in such a way that the approximation error $\|\mathbb{L} - \mathbb{L}_r\|$ is small. Many approaches for selecting sets of rows and columns have been proposed. In the following, we mention only some of them.

1. Selection based on a maximum volume sub-matrix in [124].
2. Selection based on minimizing the approximation error in the Chebyshev norm ("skeleton" approximation) in [62, 61].
3. Procedure based on the "cross-approximation" algorithm in [125].
4. Selection based on a discrete empirical interpolation method (DEIM) approach in [146].

3.5.1. The Loewner CUR algorithm

We introduce a data-driven approximation algorithm for the SISO case based on the CUR approach. This constructs a reduced-order model using an adaptive selection of the rows and columns via the cross-approximation algorithm in [125]. The steps of the procedure are included in the following algorithm.

For the practical implementation of the function "*crossapprox*," used in steps 5 and 6 of the above algorithm, we refer the reader to Algorithm 1 in [113], or to the original reference [125].

Remark 3.12 (DEIM):
Instead of using the cross approximation algorithm, one can use the DEIM (Discrete Empirical Interpolation Method) algorithm from [146]. Hence, steps 5 and 6 in Algorithm 3 need to be modified accordingly. As a result, singular value decompositions are performed to construct left and right singular vector matrices (for which the DEIM procedure is applied). To avoid the SVD, an incremental QR factorization can be used, as proposed in [146]. ◊

Algorithm 3.3: The Loewner CUR cross approximation based method

```
Input: $\mathbf{S} = [s_1, \cdots, s_N] \in \mathbb{C}^N$, $\mathbf{F} = [\phi_1, \cdots, \phi_N] \in \mathbb{C}^N$ with $N, r \in \mathbb{N}$, and
       tolerance values $\delta$, $\epsilon$.
Output: $\hat{\mathbf{E}} \in \mathbb{R}^{r\times r}, \hat{\mathbf{A}} \in \mathbb{R}^{r\times r}, \hat{\mathbf{B}} \in \mathbb{R}^{r\times 1}, \hat{\mathbf{C}} \in \mathbb{R}^{1\times r}$ with $r \ll N$.
1: Form left and right set as: $(\mu_j, \mathbf{v}_j)$, $j = 1, \ldots, q$ and $(\lambda_i, \mathbf{w}_i)$, $i = 1, \ldots, k$
2: Form the Loewner matrices $\mathbb{L}$ and $\mathbb{L}_s$ as in Algorithm 1 and step 2.
3: Transform all the complex data to real as explained in section 2.5.4.
4: $\mathfrak{J}_0 = [j_1, \cdots, j_r] \subset \mathfrak{J}_n$ an initial set of column indices.
5: $[\mathfrak{I}_r\ , \sim, \sim] = crossapprox([\mathbb{L} \quad \mathbb{L}_s], \mathfrak{J}_0, \delta, \epsilon)$.
6: $[\sim, \mathfrak{J}_r\ , \sim] = crossapprox(\begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix}, \mathfrak{I}_r, \delta, \epsilon)$.
6: $\hat{\mathbf{E}} = -\mathbb{L}(\mathfrak{I}_r, \mathfrak{J}_r)$, $\hat{\mathbf{A}} = -\mathbb{L}_s(\mathfrak{I}_r, \mathfrak{J}_r)$, $\hat{\mathbf{B}} = \mathbb{V}(\mathfrak{I}_r)$, $\hat{\mathbf{C}} = \mathbb{W}(\mathfrak{J}_r)$.
7: The rational approximant is given by:
       $$\mathbf{H}_r(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}}.$$
```

Remark 3.13 (Data preservation):
The CUR factorization directly reveals the dominant rows/columns of the data, while the SVD does not. More precisely, the leading singular vectors give only linear combinations of the underlying features. Meanwhile, with the CUR, one gets an actual subset of the initial features (columns) together with the corresponding rows. Consequently, the first benefit of the CUR is that it preserves the physical meaning and structure of the initial data. Additionally, another advantage is that the sparsity is preserved. ◊

3.5.2. Choice of left and right interpolation points

This section deals with the problem of selecting the initial interpolation points in the Loewner framework. More specifically, we investigate how the choice of the initial interpolation points affects the quality of the reduced-order model. We take into consideration different point distributions in 1D or 2D.

Moreover, several splitting techniques are analyzed. These are related to partitioning the data set into two disjoint subsets performed at the beginning of the algorithms in the Loewner framework.

3.5.3. Distribution of the interpolation points

We present the initial interpolation point distributions for the one-dimensional case (1D) and the two-dimensional case (2D).

1D interpolation grid	2D interpolation grid
equispaced;	equispaced ("same areas");
logarithmic spaced;	logarithmic spaced;
Chebyshev nodes;	Padua points;
Uniformly random;	Uniformly random;

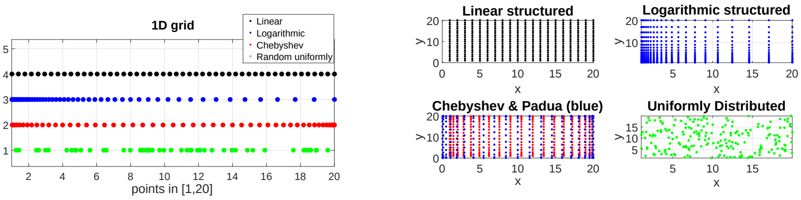


Figure 3.4.: A visual representation of different interpolation grids

In Fig. 3.4, we depict different distributions of initial interpolation points. One way of selecting points is equispaced or linearly spaced, commonly used for Fourier analysis. This represents a natural choice because of the usage of trigonometric periodic functions. In some practical applications, the choice of logarithmically distributed points is more appropriate when the energy decreases exponentially as time or frequency approaches infinity (on an unbounded domain).

Naturally, a dense sampling grid can be used at the beginning of the experiment (e.g., for a lower frequency range or small-time instances). The motivation for this approach stems from the assumption that meaningful quantities (with high energy or with relevant oscillations) appear in the beginning, requiring more samples. Afterward, a more sparse distribution grid of points can be chosen instead as the energy level decays (or as relevant oscillations decay in time).

Additionally, the choice of Chebyshev-type points is motivated by their usage in polynomial-based interpolation on bounded domains due to, for example, the elimination of the Runge phenomenon[3] (high degree polynomials are generally unsuitable for interpolation with equispaced points). Finally, randomly distributed sampling points often appear in stochastic experiments characterized by randomness.

3.5.4. Partition of the data points and values

Data splitting is one of the first steps in the classical Loewner algorithm (presented in Section 2). In this section, we mention various splitting schemes and how they affect the Loewner matrix singular value decay and the approximation quality of the Loewner interpolants.

The data set ($n =$ even) is composed of the following:

$$\begin{cases} \text{Sample points}: \ \mathbf{S} = [\omega_1, \omega_2, \cdots, \omega_n] \in \mathbb{R}^n, \ \text{with } \omega_1 < \omega_2 < \ldots < \omega_n, \\ \text{Sample values}: \ \mathbf{H} = [H(\omega_1), H(\omega_2), \cdots, H(\omega_n)] \in \mathbb{C}^n. \end{cases} \tag{3.18}$$

[3] Runge's phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of a high degree over a set of equispaced interpolation points

We analyze four types of data splitting mentioned in the following.

1. **First type**: disjoint splitting.
   - $\boldsymbol{\mu} = [\omega_1, \cdots, \omega_{n/2}]$ and $\mathbb{V} = [H(\omega_1), \cdots, H(\omega_{n/2})]$,
   - $\boldsymbol{\lambda} = [\omega_{n/2+1}, \cdots, \omega_n]$ and $\mathbb{W} = [H(\omega_{n/2+1}), \cdots, H(\omega_n)]$.
2. **Second type**: alternate splitting.
   - $\boldsymbol{\mu} = [\omega_1, \omega_3, \cdots, \omega_{n-1}]$ and $\mathbb{V} = [H(\omega_1), H(\omega_3), \cdots, H(\omega_{n-1})]$,
   - $\boldsymbol{\lambda} = [\omega_2, \omega_4, \cdots, \omega_n]$ and $\mathbb{W} = [H(\omega_2), H(\omega_4), \cdots, H(\omega_n)]$.
3. **Third type**: magnitude splitting (in this case the set **S** is first sorted with respect to the magnitude of set **H**).
   - $\boldsymbol{\mu} = [\omega_1, \cdots, \omega_{n/2}]$ and $\mathbb{V} = [H(\omega_1), \cdots, H(\omega_{n/2})]$,
   - $\boldsymbol{\lambda} = [\omega_{n/2+1}, \cdots, \omega_n]$ and $\mathbb{W} = [H(\omega_{n/2+1}), \cdots, H(\omega_n)]$.
4. **Forth type**: magnitude alternate splitting (in this case, the set **S** is first sorted with respect to the magnitude of set **H** and then alternating splitting is applied).
   - $\boldsymbol{\mu} = [\omega_1, \omega_3, \cdots, \omega_{n-1}]$ and $\mathbb{V} = [H(\omega_1), H(\omega_3), \cdots, H(\omega_{n-1})]$,
   - $\boldsymbol{\lambda} = [\omega_2, \omega_4, \cdots, \omega_n]$ and $\mathbb{W} = [H(\omega_2), H(\omega_4), \cdots, H(\omega_n)]$.

As observed in practice, when splitting the data for the first type, the Loewner matrix has a very fast decay of the singular values. Moreover, the computed reduced models usually provide low approximation quality.

On the other hand, for the second separation type (alternate splitting), the left and right sets of sample points can be chosen ϵ-close to one another (element-wise). Hence, as $\epsilon \to 0$, Hermitian interpolation conditions are enforced (which involve matching the first derivative at those points).

Another observation in the case of second-type splitting is that the numerical rank of the Loewner matrix is usually larger than that of the Loewner matrix constructed based on the first type. Additionally, for the second type, the condition number is smaller than that computed for the first type. For the above-mentioned cases, bounds on the singular value decay of the Loewner matrix are provided in [21].

3.6. Numerical examples

This section illustrates the concepts developed in the preceding sections and chapters employing examples. In particular, the following seven examples will be analyzed.

1. The benchmark CD player ($n = 120$).
2. The function $f(x) = exp(-x)\sin(10x),\ \ x \in [-1, 1]$.
3. The inverse of the Bessel function of the first kind, in $[0, 10] \times [-1, 1]$j.

4. An Euler-Bernouli beam described by a PDE.

5. A heat equation with transfer function $\mathbf{H}(s) = exp(-\sqrt{s}),\ s \in [0.01, 100]\text{j}$.

6. Approximation of $f = y/\sinh(y)$, $y(x) = 100\pi(x^2 - 0.36),\ x \in [-1, 1]$.

7. The *sign* function sampled on the reunion of the interval $[-b, -a]$ and $[a, b]$, $a > b > 0$.

3.6.1. Approximation of the CD player

Consider the CD player benchmark example, a MIMO dynamical system of dimension 120 with 2 inputs and 2 outputs. Here we will consider the $(2, 1)$ sub-system, i.e., the SISO system from the first input to the second output.

The goal is to approximate the transfer function in the Loewner framework. We start by considering 400 interpolation points $\pm\text{j}\omega_i$, $i = 1, \cdots, 200$, where ω_i are logarithmically spaced in the interval $\Omega = [10^{-1}, 10^5]$. Thus $\Omega = \{\omega_1, \omega_2, \cdots, \omega_{200}\}$, where $\omega_i < \omega_{i+1}$, for all i. We now define the left/right interpolation points in four ways, as explained in section 3.3.2 and depicted in Fig. 3.5 (up).

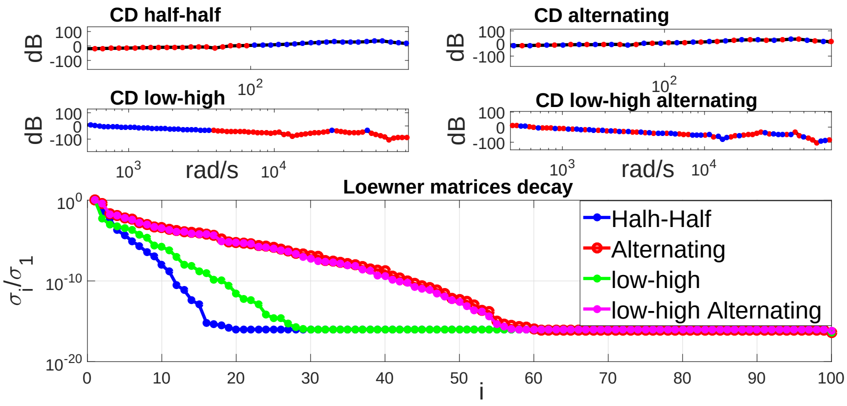


Figure 3.5.: The four different splitting schemes (up) and the decay of the singular values $(\frac{\sigma_i}{\sigma_1},\ i = 1, \ldots, 100)$ of the Loewner matrix for each type (down).

As it can be seen in Fig. 3.5 (down), the decay of the Loewner matrix singular values is faster for "half-half" (disjoint) splitting than for "alternating" splitting.

The next step is to choose the truncation order and determine the level of approximation. We propose two different ways for this purpose.

1. By choosing equal truncation orders r.

2. By choosing for each separation the maximum truncation order so that $\frac{\sigma_r}{\sigma_1}$, is equal to a fixed tolerance value.

First experiment: equal truncation orders. Here, we fix the truncation order to $r = 10$ and compute $\frac{\sigma_r}{\sigma_1}$. The results are presented in the table that follows. The frequency response of the original system with those of the four reduced systems (corresponding to each different splitting) is shown in Fig. 3.6. Note that all methods produce good approximation quality.

case	1st	2nd	3rd	4th
r	10	10	10	10
$\frac{\sigma_r}{\sigma_1}$	$1e-8$	$1e-6$	$1e-4$	$1e-4$

Table 3.2.: Normalized singular values corresponding to $r = 10$ for each splitting.

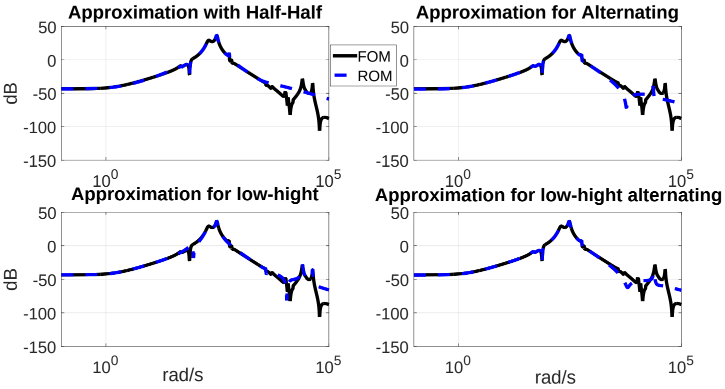


Figure 3.6.: Frequency response comparison: original system vs. the reduced ones with equal truncation orders ($r = 10$).

Second experiment: reaching machine precision The tolerance of normalized singular value $\frac{\sigma_r}{\sigma_1}$ is now fixed (e.g. 10^{-14}). This implies the truncation order r. The results are presented in the following table. The truncation order for the first splitting type is more than three times smaller than that for the second splitting type (16 vs. 51). The frequency response of the original systems with the four reduced systems is depicted in Fig. 3.7. All methods produce good approximation quality, with a slight deviation in the high-frequency range observed for the first splitting type.

case	1st	2nd	3rd	4th
r	16	51	23	48
$\frac{\sigma_r}{\sigma_1}$	$1e-14$	$1e-14$	$1e-14$	$1e-14$

Table 3.3.: Different truncation orders for all splitting schemes and a fixed tolerance.

Finally, Fig. 3.8 shows the approximation error for each reduced system.

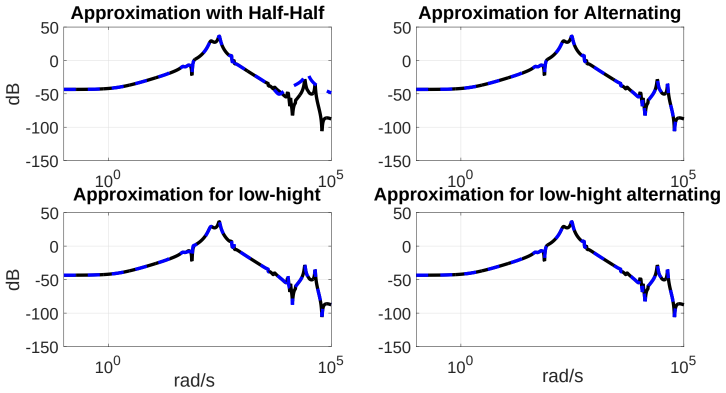


Figure 3.7.: Frequency response comparison: original system vs. the reduced ones by reaching machine precision.

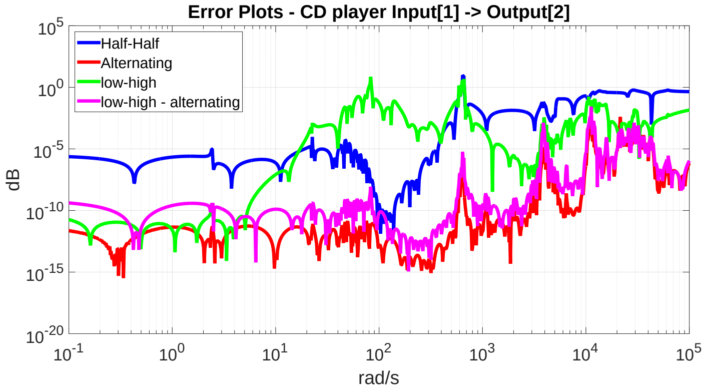


Figure 3.8.: Approximation error for the four splitting schemes.

Notice that the blue curve in Fig. 3.8 has a 'V' shape in the middle of the sampling interval. The lowest approximation error is recorded for the second splitting type (alternate selection).

3.6.2. Approximation of an oscillating function

We collect $N = 4,000$ measurements $\{(s_k, \phi_k) : k = 1, ..., N\}$ of the following function:

$$\phi(x) = e^{-x}\sin(10x), \quad x \in [-1, 1]. \tag{3.19}$$

Assume that the interpolation points $\mathbf{s} = [s_1, s_2, ..., s_{4000}] \subset [-1, 1]$ are *equispaced*; Next, we have two types of splitting.

1. **First type**: disjoint splitting.
   - Left: $\boldsymbol{\mu} = [s_1, s_2, ..., s_{2000}] \subset [-1, 0)$
   - Right: $\boldsymbol{\lambda} = [s_{2001}, s_{2002}, ..., s_{4000}] \subset [0, 1]$

We construct the Loewner pencil, and the underlying rank is 11.

2. **Second type**: alternate splitting.
    - Left: $\boldsymbol{\mu} = [s_1, s_3, ..., s_{3999}] \subset [-1, 1]$
    - Right: $\boldsymbol{\lambda} = [s_2, s_4, ..., s_{4000}] \subset [-1, 1]$.

    We construct the Loewner pencil, and the underlying rank is 15.

Fig. 3.9 shows the entries of the Loewner matrix in logarithmic scale for the two ways of sampling point separation. Next, the interpolation data is compressed, making use of the following methods: (a) the singular value decomposition SVD, (b) the randomized version rSVD, (c) CUR, implemented with DEIM, and (d) CUR implemented with cross approximation. The parameters for the latter two methods are $\epsilon = 0.001$ and $\delta = 0.01$. The error curves for the first splitting are shown in Fig. 3.10. The red Xs indicate the selected points with the CUR-cross approximation method, while the green crosses (+) indicate the selected points with the CUR-DEIM method. The error curves for the second splitting are shown in Fig. 3.11. As opposed to the previously shown results (in Fig. 3.10), the error in this case (Fig. 3.11) is distributed more uniformly.

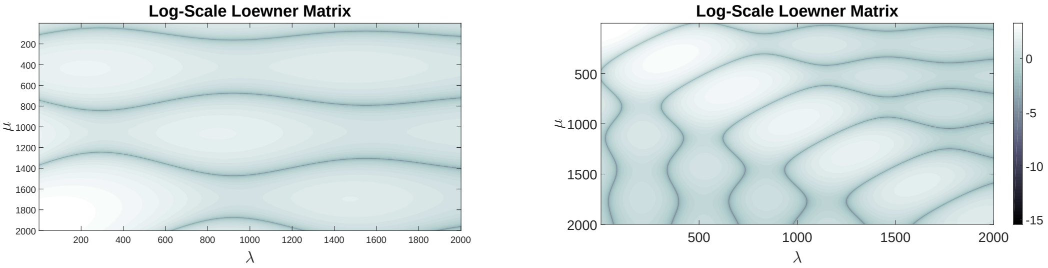


Figure 3.9.: Entries of the Loewner matrix for the first splitting (left) and the second splitting (right).

Reduction - r **for** $\approx \mathbb{L}$	$rank(\mathbb{L}_{r\times r})$	$cond(\mathbb{L}_{r\times r}) = \frac{\sigma_{max}}{\sigma_{min}}$	**Error** $\|\cdot\|_{\mathbf{F}}$	$Time\ (s)$
SVD	11	$9.7313e+10$	$6.7367e-10$	4.166029
CUR-CrossApprox	11	$7.6582e+10$	$1.5621e-09$	0.528352
CUR-DEIM	11	$1.3898e+11$	$2.2283e-09$	4.101303
randomized SVD	11	$9.7314e+10$	$1.1281e-10$	0.030148

Table 3.4.: Results for the first splitting type (disjoint) with an i5-CPU 2.60 GHz.

As seen in the above experiments, the data splitting influences the Loewner singular value decay and the approximation quality. In most of the following experiments, we choose the "alternating" way of splitting the data.

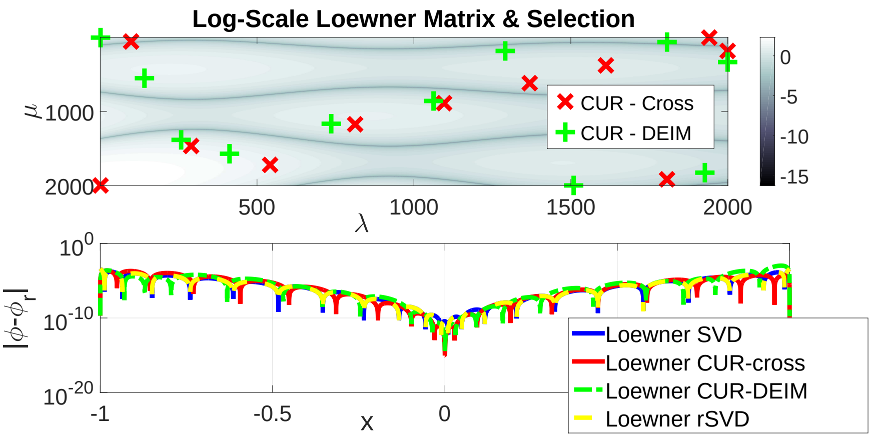


Figure 3.10.: Selected points and approximation error for the disjoint splitting

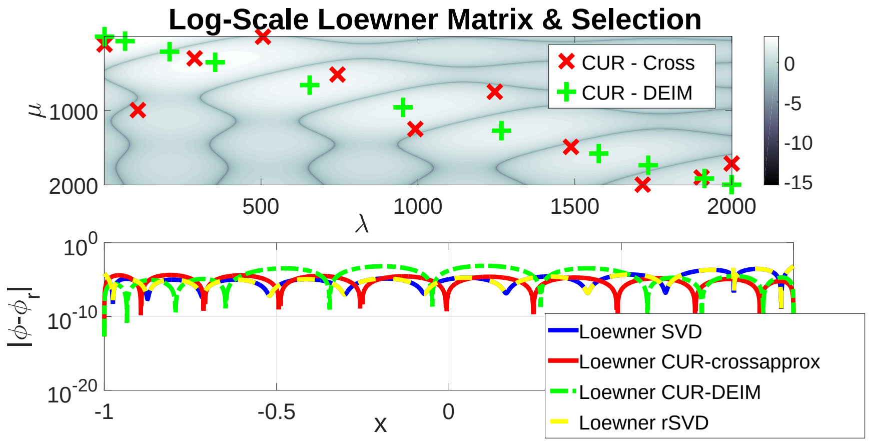


Figure 3.11.: Selected points and approximation error for the alternate splitting.

Reduction - r for $\approx \mathbb{L}$	$rank(\mathbb{L}_{r\times r})$	$cond(\mathbb{L}_{r\times r}) = \frac{\sigma_{max}}{\sigma_{min}}$	**Error** $\|\cdot\|_{\mathbf{F}}$	$Time(s)$
SVD	11	$8.8199e+4$	0.0020	4.261075
CUR-CrossApprox	11	$1.0228e+5$	0.0062	0.563411
CUR-DEIM	11	$9.3343e+4$	0.0245	4.152420
randomized SVD	11	$8.8199e+4$	0.0020	0.024586

Table 3.5.: Results for the second splitting type (alternate) with an i5-CPU 2.60 GHz.

3.6.3. Approximation of a Bessel function

This section investigates the approximation of a Bessel function's inverse in a complex plane domain. If this function is considered to be the transfer function of a dynamical system, this system is infinite-dimensional; furthermore, it is not stable as there are poles in the right half of the complex plane. In particular, we consider the inverse of the Bessel function of the first kind and order $n \in \mathbb{N}$. The following contour integral defines it:

$$J_n(s) = \frac{1}{2\pi i} \oint e^{\left(\frac{s}{2}\right)\left(t-\frac{1}{t}\right)} t^{-n-1} dt. \tag{3.20}$$

Here, we consider only the case $n = 0$. Our aim is to approximate $H(s) = \frac{1}{J_0(s)}$, $s \in \mathbb{C}$, inside the rectangle $\mathbf{\Omega} = [0, 10] \times [-1, 1] \subset \mathbb{C}$. In Fig. 3.12 (left pane), the function $H(s)$ is shown in the domain $\mathbf{\Omega}$. The three spikes correspond to the unstable poles of the underlying system. These are three of the zeros of the Bessel function. Here we construct approximants $H_r(s)$, of order r, of $H(s)$, using the interpolation points as shown in Fig. 3.12 on the right pane. The distribution of the two-dimensional initial grids is $5,000$ Padua points with the conjugates. This grid is used to reduce the Runge phenomenon. For more details in approximation theory (i.e., Runge phenomenon, Padua points, barycentric interpolation, etc.), we refer the reader to [149]. In [100, 108], the same experiment is presented with other grids (randomly uniformly structured) types.

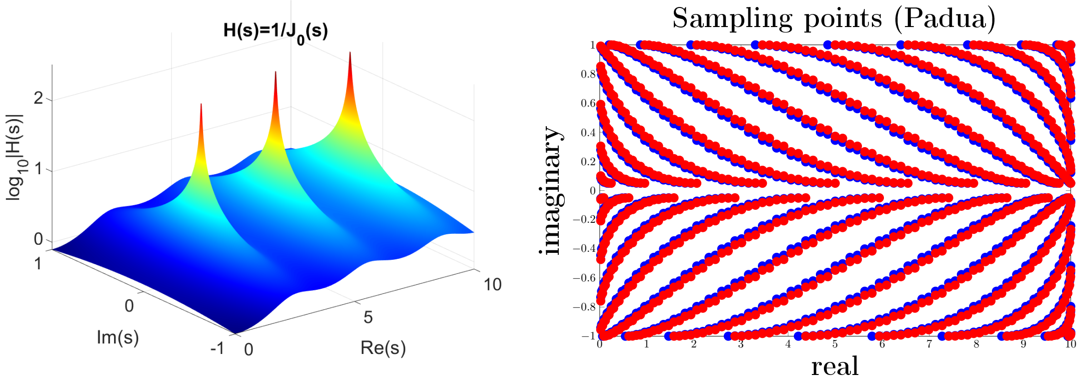


Figure 3.12.: Left: The inverse of the Bessel function of the 1st kind. Right: A subset of $10,000$ Padua point grid over $\mathbf{\Omega} = [0, 10] \times [-1, 1]$ domain are shown.

In the Loewner framework, the singular value decomposition (SVD) plays a key role. This factorization allows us to extract the numerical order of the rational model, which approximates the original non-rational one.

In Fig. 3.13 (left pane), we show the distribution of the normalized singular values $\frac{\sigma_j}{\sigma_1}$, $j = 1, ..., N$, of the augmented matrices $[\mathbb{L} \quad \mathbb{L}_s]$ and $\begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix}$.

By taking measurements as in Fig. 3.12 (right pane), the decay of the singular values Fig. 3.13 - left pane, leads to a reduced order $r = 12$ with $\frac{\sigma_{12}}{\sigma_1} = 4.887 \cdot 10^{-13}$. In Fig. 3.13 on the right pane, the pole/zero diagrams are presented, which include the results from all methods. Methods VF and Loewner(SVD or CUR) construct real strictly rational models with degree $(11, 12)$[4] with $D = 0$, as opposed to the AAA algorithm, which constructs a complex proper rational model of degree $(12, 12)$ with a non-zero D term.

[4] Notation (m, n) indicates that the order of the numerator polynomial is m. The order of the denominator polynomial is n.

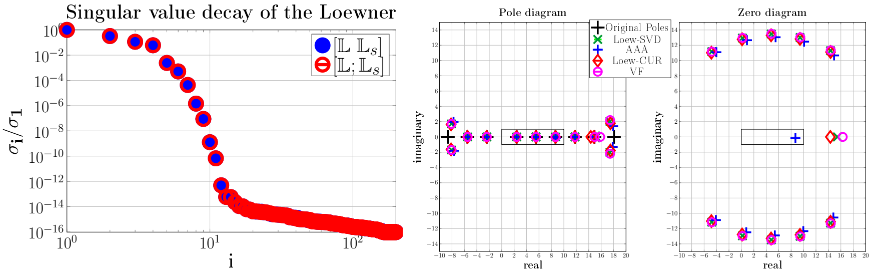


Figure 3.13.: Left: Singular value decay of 10, 000 values. Right: Pole/zero diagrams with the three original poles (zeros of Bessel) were recovered with 15 digits of accuracy.

Points from the sampling grid are selected using LoewCUR-cross and AAA methods. Applying the LoewSVD method, the point selection is obtained by compressing the initial grid. This can be achieved by using the first r columns (r : singular vectors) of the singular matrices as projection matrices and by solving two ($r\times r$-dimension) generalized eigenvalue problems as explained in section 3.2.5. In this way, we compress the original grid with $N = 10,000$ points into a much smaller set of only $2r = 24$ points which are exact interpolation points for the approximant. As it turns out, the projected points lie in the domain $\mathbf{\Omega}$; see also left pane in Fig. 3.14.

The LoewCUR-cross and AAA methods select points among the initial interpolation points but with different criteria. The AAA algorithm selects support points by minimizing the mean squared error with the rest of the measurements. At the same time, LoewCUR uses cross approximation, which maximizes the absolute value of the determinant (maximum volume) of the sub-matrix of dimension ($r \times r$).

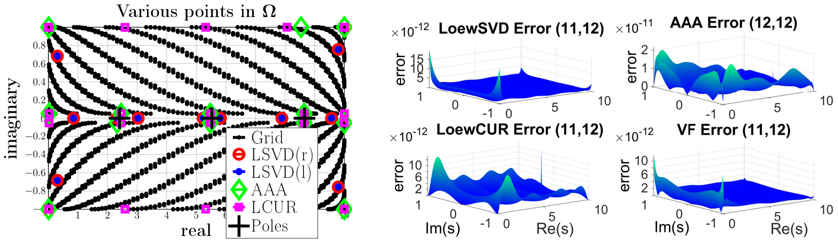


Figure 3.14.: Left: Support and compressed points for every method over $\mathbf{\Omega}$ domain with LSVD(r) → LoewSVD projected right points, LSVD(l) → LoewSVD projected left points. Right: The error for every method.

In Fig. 3.14 on the right pane, the error for each method is shown. The normalized error is computed as: $\frac{|H(s)-H_r(s)|}{|H(s)|}$ with 25, 000 evaluation points in $\mathbf{\Omega}$. It should be mentioned that the above special choice of the original interpolation grid as Padua points indeed reduced the Runge phenomenon.

Next, we wish to visualize the approximation error outside $\mathbf{\Omega}$. Towards this goal we chose 25, 000 equispaced evaluation points inside the domain $[-3, 13] \times [-3, 3]$. Results with log-contour level error of increasing order $10^{-16}, ..., 10^{-4}$ are presented in Fig. 3.15.

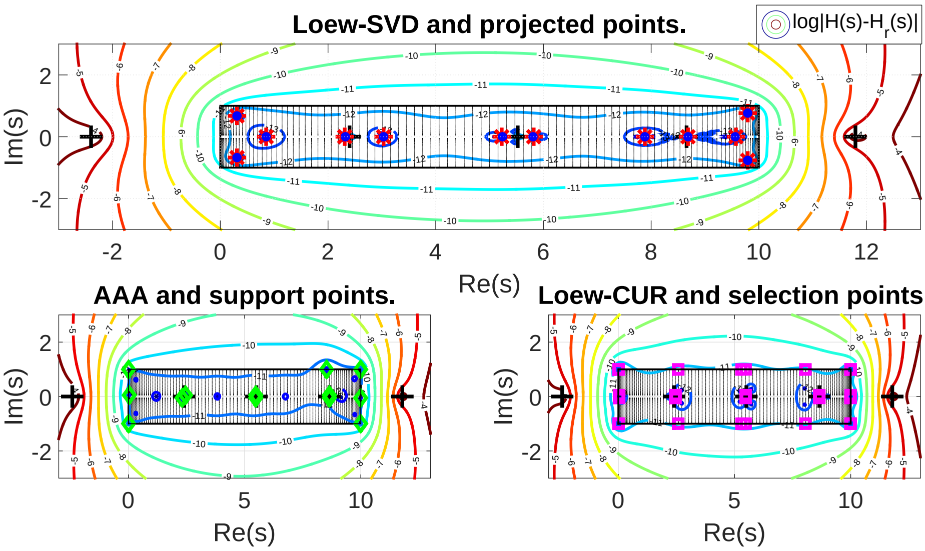


Figure 3.15.: Extrapolation error as $\log|H(s) - H_r(s)|$ in $[-3, 13] \times [-3, 3]\mathrm{j} \subset \mathbb{C}$. With '+' the original poles.

All methods constructed accurate, rational approximations. Notice, however, that the Loewner approach reaches similar precision with AAA without performing any optimization step. Finally, the CUR method performed the best in terms of computational complexity.

3.6.4. Approximation of an Euler Bernoulli beam

In this subsection, we analyze the approximation of an Euler-Bernoulli clamped beam [47] and study in [67, 158]. The underlying PDE describes the oscillation of the free end. As shown in [47], the non-rational transfer function is given by:

$$\begin{aligned} H(s) &= \frac{sn(s)}{(EI + sc_d I)m^3(s)d(s)}, \text{ where} \\ m(s) &= \left[\frac{-s^2}{EI + c_d I s}\right]^{\frac{1}{4}}, \ d(s) = 1 + \cosh(Lm(s))\cos(Lm(s)), \\ n(s) &= \cosh(Lm(s))\sin(Lm(s)) - \sinh(Lm(s))\cos(Lm(s)). \end{aligned} \tag{3.21}$$

Usually, the next step consists of a discretization of the PDE involved. Instead, we bypass this step and take frequency response measurements using the transfer function above. The parameter specification is as in [47][5]. Thus, we have the frequency response of the beam as in Fig. 3.16 and on the left pane.

[5] *Young's modulus (elasticity constant):* $E = 69\ GPa = 6,9 \cdot 10^{10} N/m^2$*, moment of inertia:* $I = 3.58 \cdot 10^{-9} m^4$*, damping constant:* $c_d = 5 \cdot 10^{-4}$*, length:* $L = 0.7m$*, base:* $b = 0.07m$*, height:* $h = 0.0085m$*.*

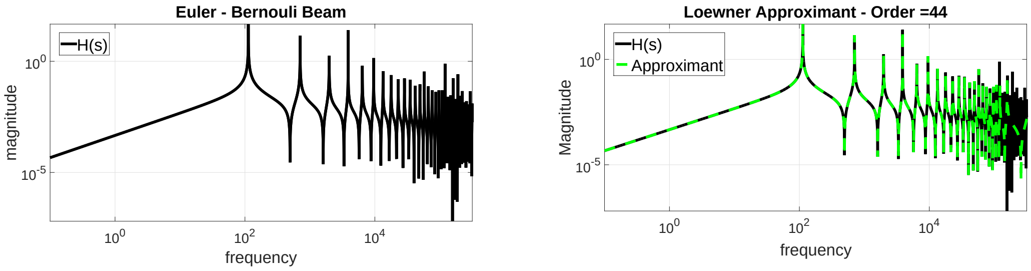


Figure 3.16.: Left: Original frequency response of the beam. Right: The approximant is constructed with the Loewner framework.

The next step is to collect 2,000 measurements on the imaginary axis (frequencies $j\omega_i, i = 1, ..., 2000$), spaced logarithmically from 1 rad/s to 10^5 rad/s. These points are depicted in the left pane of Fig. 3.17.

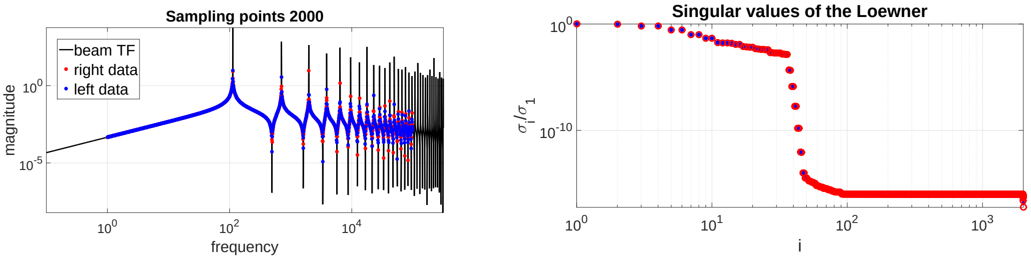


Figure 3.17.: Left: 2,000 sampling points alternating as left and right. Right: The singular value decay.

The Loewner matrices decay's singular value is shown in Fig. 3.17 on the right pane. Thus, we construct a reduced model with dimension $r = 44$, and the Loewner approximant in Fig. 3.16 (right pane) is depicted.

Finally, the poles and zeros for every method are presented in Fig. 3.18. Each method's approximation quality is given in Fig. 3.19, where the evaluation is in the frequency range from 1 to $10^{5.5}$. The error outside the sampling domain increases, thus indicating the difficulty of approximation outside the sampling domain for infinite dimensional systems.

3.6.5. Approximation of the heat equation

Next, we investigate a one-dimensional heat equation [18]. The corresponding PDE describing the diffusion of heat leads to the following non-rational transfer function:

$$H(s) = e^{-\sqrt{s}}, \ s \in \mathbb{C}. \tag{3.22}$$

The aim is to construct reduced models using the Loewner framework and compare the results with the TF-IRKA used in [18]. Iterative Rational Krylov Algorithm - IRKA [19] builds optimal reduced models by minimizing the $\mathcal{H}_2$ error [87].

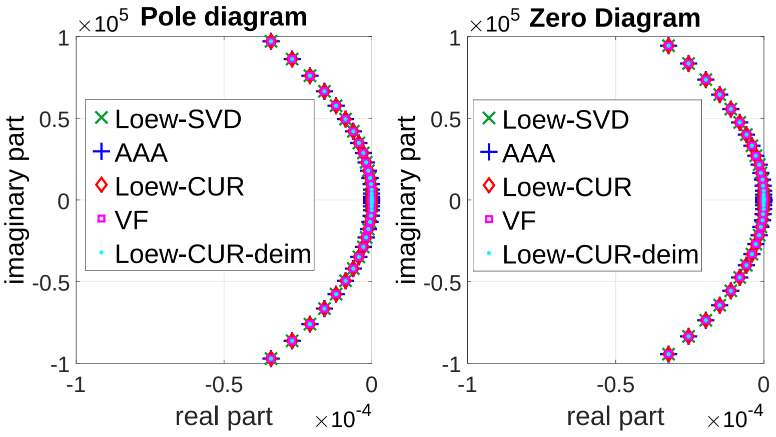


Figure 3.18.: Pole/Zero diagram for every method (LoewSVD, LoewCUR-cross, LoewCUR-DEIM, VF, and AAA).

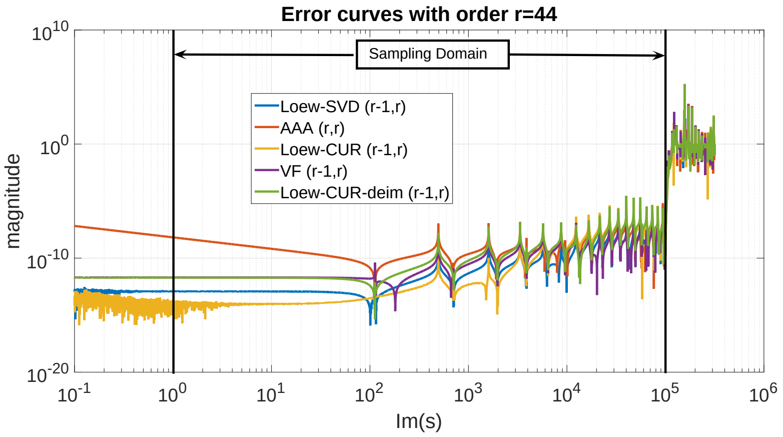


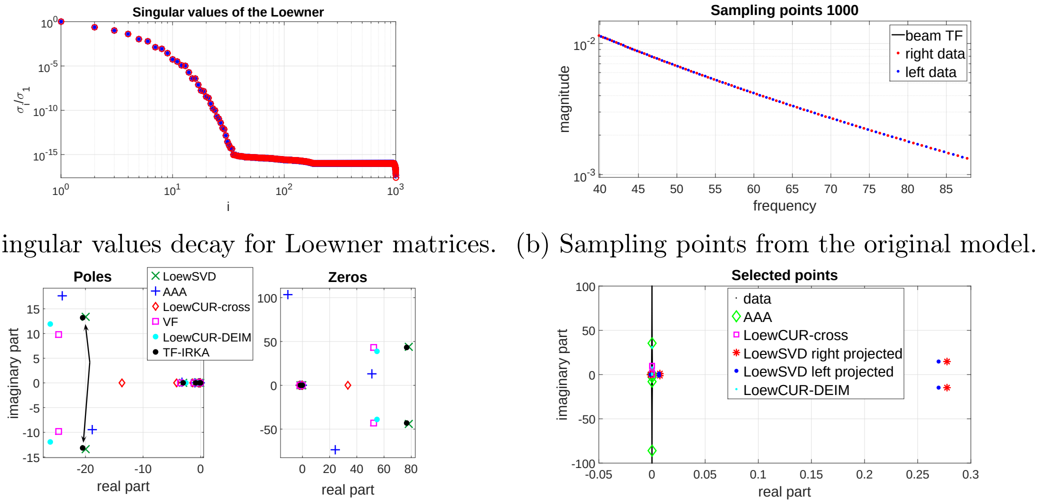
Figure 3.19.: The error distribution with 8, 000 evaluation points grid.

By collecting 1, 000 values of the transfer function on the imaginary axis, the resulting reduced order was $r = 6$ (as in [18]). For this truncation order, it holds: $\frac{\sigma_6}{\sigma_1} \approx 6 \cdot 10^{-3}$. In Fig. 3.20c, the pole/zero distribution for every method is depicted; in Fig. 3.20d, the selected points are shown. It is worth mentioning that the Loewner SVD method produced poles near the optimal set computed using IRKA; see Fig. 3.20c. Approximation results in Fig. 3.21.

3.6.6. Approximation of a two-peak function

This section presents an example of a hyperbolic sine from [123]. The difficulty here results from the two differentiable peaks. More precisely, the function is

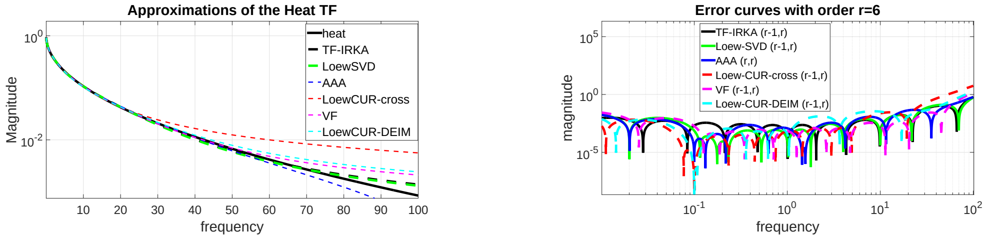
$$f(x) = \frac{100\pi(x^2 - 0.36)}{\sinh(100\pi(x^2 - 0.36))}, \quad x \in [-1, 1],$$



(a) Singular values decay for Loewner matrices. (b) Sampling points from the original model.

(c) Pole/Zero diagram for every method. (d) Projected/selected points.

Figure 3.20.: Approximation of the heat equation with LoewSVD, LoewCUR, VF, AAA, TF-IRKA.



(a) Original transfer function superimposed with the low order approximants.

(b) Error profile for every method also included the optimal.

Figure 3.21.: Approximation results for the heat equation with various interpolation methods.

and is shown Fig. 3.22 (left pane). We approximate this function by choosing 1,000 equispaced points in $[-1,1]$ as on the right pane in Fig. 3.22. The singular values of the Loewner matrix are shown in Fig. 3.23 on the left pane, while the selected points are shown on the right pane of the same figure. The order is selected to be $r = 38$ with $(\frac{\sigma_{38}}{\sigma_1} \approx 10^{-12})$. In Fig. 3.24, each method's distribution of the poles and zeros is shown. On the other hand, AAA looks quite different because it doesn't impose real symmetry.

Remark 3.14 (Imaginary points):
The different support points are shown in Fig. 3.23, right pane. In the case of the LoewSVD method, two almost pure imaginary projected points are obtained even if the initial sampling points were real. ◊

Finally, we observe a good fit for every method, with slightly better performance attained

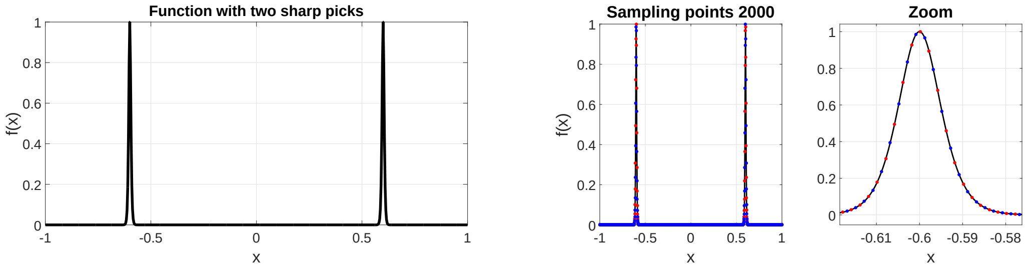


Figure 3.22.: Left: The function f with two very sharp differentiable picks. Right: 1,000 sampling points and zoom in close to one pick.

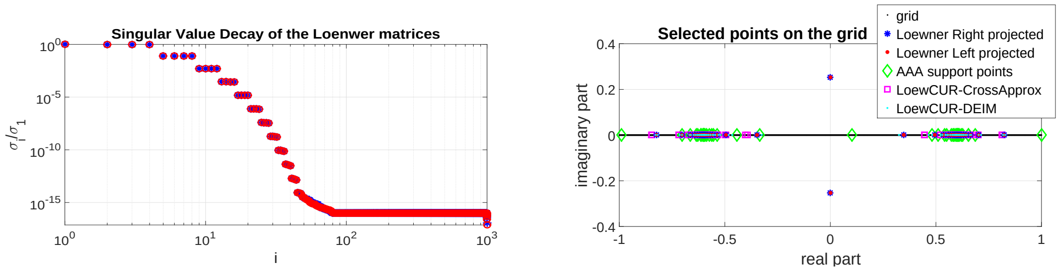


Figure 3.23.: Left: Singular values decay. Right: Various points for every method and the projected points from the Loewner framework.

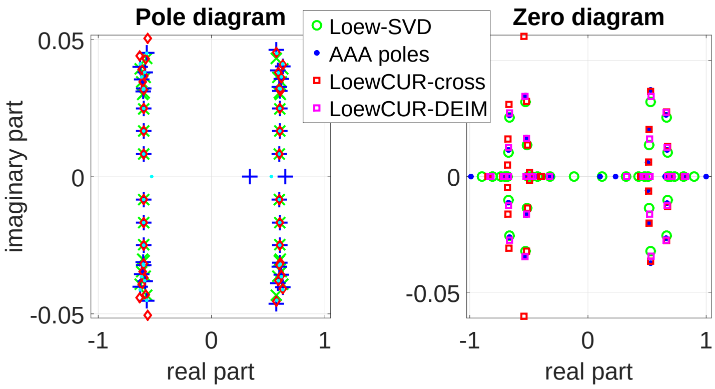


Figure 3.24.: The pole/zero diagram.

for the Loewner SVD method (see the error plot in Fig. 3.25).

3.6.7. Approximation of the sign function

Our final case study problem concerns the approximation of the *sign* function, known as Zolotarev's 4th problem. Here, we compare the approximation obtained using the Loewner SVD with the optimal solution that is explicitly known [2]. Given two disjoint closed complex sets E and F, Zolotarev's 4th problem is to find the rational function

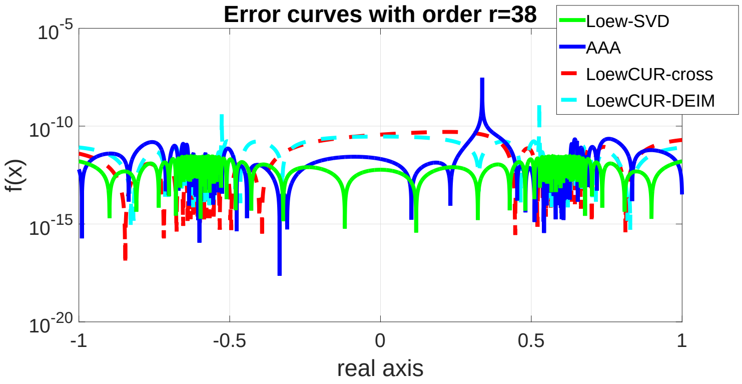


Figure 3.25.: The error profile with 5, 000 evaluation points over $[-1, 1]$.

$r(x) = \frac{p(x)}{q(x)}$, where p, q are polynomials of degree k, that deviates least from the sign function

$$sign(x) = \begin{cases} -1, & x \in E, \\ +1, & x \in F, \end{cases}$$

on $E \cup F$. For general sets E and F, the solution to Zolotarev's 4th problem is not known. However, there are special cases where the rational function can be given explicitly. For the real disjoint intervals, $E = [-b, -1]$, and $F = [1, b]$ with $b > 1$, an explicit (optimal) solution to Zolotarev's fourth problem is known [2]. Here, we investigated how well the Loewner framework can approximate this discontinuous function in two symmetric real intervals. We choose $b = 3$ and $N = 2,000$ initial interpolation points from $[-3, -1] \cup [1, 3]$. We perform two experiments. First, we choose initial interpolation points as equispaced and, secondly, as Chebyshev nodes. We split the data for each choice as "half-half" and "alternating," as discussed previously. The left pane in Fig. 3.26 shows the plot of the $sign$ function. In [34], the explicit solution of this optimization problem is

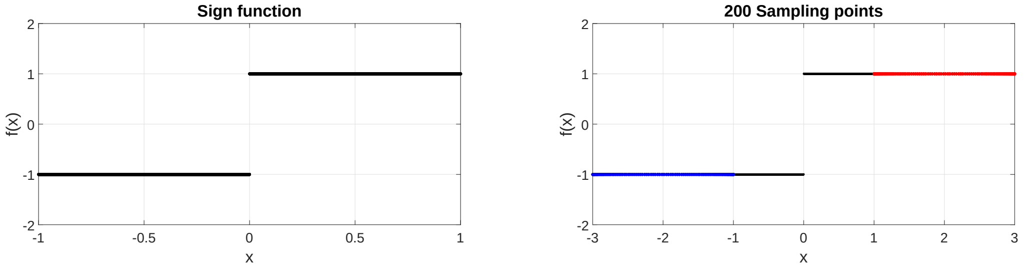


Figure 3.26.: Left: The $sign$ function. Right: Chebyshev nodes in $[-3, -1] \cup [1, 3]$.

computed. We start by taking $N = 2,000$ measurements as Chebyshev nodes as in Fig. 3.26 on the right pane. The above sampling way leads to the following singular value decay of the Loewner matrices as in Fig. 3.27 on the left pane. From the rank revealing factorization in the left pane in Fig. 3.27, we chose $r = 4$ with $\frac{\sigma_4}{\sigma_1} = 1.657 \cdot 10^{-4}$. In Fig. 3.27 on the right pane, the pole/zero diagrams distribution derived from the Loewner

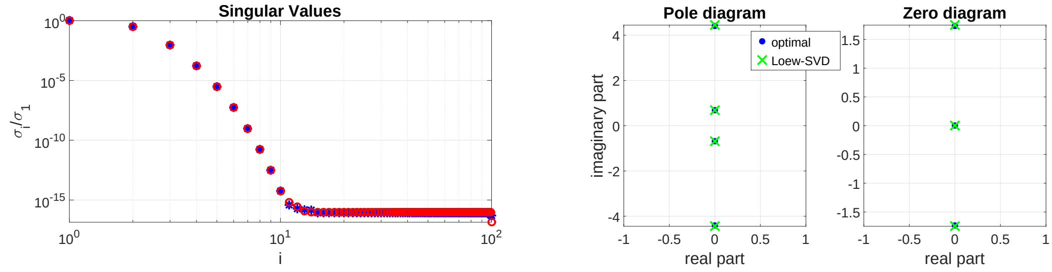


Figure 3.27.: Left: The singular value decay of the Loewner pencil. Right: Pole/Zero diagram for the Loewner and the optimal approximant with order $r = 4$.

SVD method is compared to the optimal set. In Fig. 3.28 (left) the Loewner approximant is shown. It is close to the optimal one by choosing the Chebyshev nodes and splitting the left and right points as "half-half." Indeed, the error distribution as presented in the optimal interpolant in Fig. 3.29 with the blue line has the equioscillation property of the optimal approximant in the *infinity norm* - $\|\mathbf{x}\|_\infty = max\,(|x_1|, \ldots, |x_n|)$. Thus the equioscillation of the error $|sign(x) - r(x)|$ on both intervals shows the optimality of the approximant. The Loewner framework succeeds in constructing an approximant very close to the optimal. Another aspect is shown in Fig. 3.28 (right pane). More specifically, note that the projected points are indeed interpolation points.

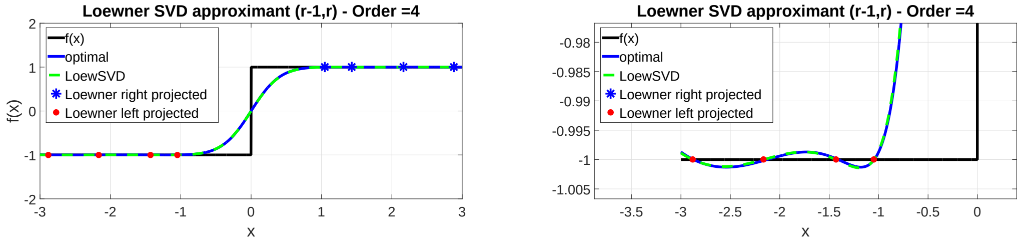


Figure 3.28.: Left: A comparison between the Loewner approximant with the optimal one order $r = 4$. Right: The projected points are approximated interpolation points.

Remark 3.15 (Interpolation goals):
If the choice of the splitting is disjoint - "half-half" as in this experiment, the constructed approximant interpolates the data as in Fig. 3.28(right pane). If the choice is "alternating" by mixing left and right, then the projected low-order model approximates the values and the derivatives at the interpolation points Fig. 3.30. $\Diamond$

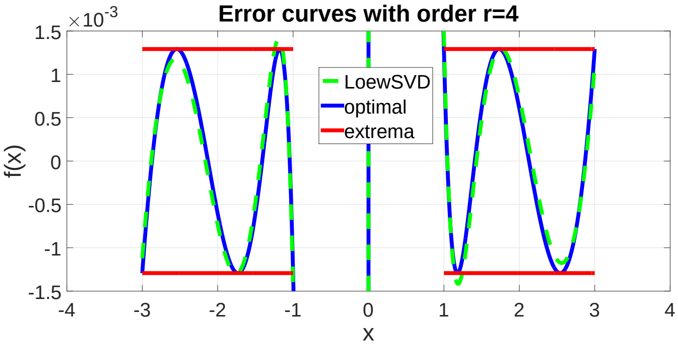


Figure 3.29.: Error plot with the Loewner approximant and the optimal solution with order $r = 4$.

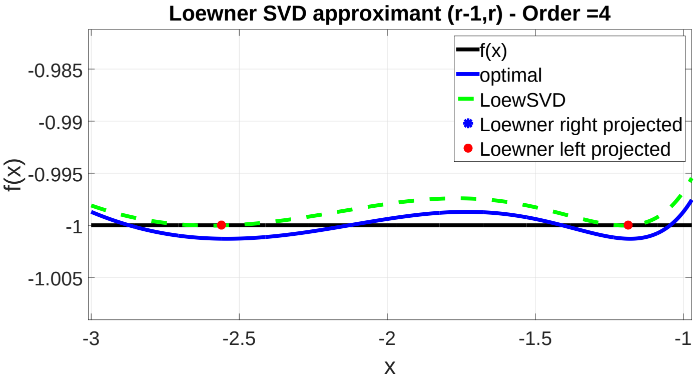


Figure 3.30.: By splitting the data as "alternating," the projected Loewner model approximates the first derivative as well (*Hermitian interpolation conditions*).

3.7. Linear systems with initial conditions

Consider the following linear system with inhomogeneous initial conditions

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \;\; \mathbf{x}(0) = \mathbf{x}_0 \\ \;\;\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \;\; t \geq 0. \end{cases} \tag{3.23}$$

By performing a Laplace transform, we can write

$$\begin{cases} s\mathbf{E}\hat{\mathbf{x}}(s) - \mathbf{E}\mathbf{x}_0 = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}\hat{\mathbf{u}}(s), \\ \qquad\qquad \hat{\mathbf{y}}(s) = \mathbf{C}\hat{\mathbf{x}}(s) + \mathbf{D}\hat{\mathbf{u}}(s), \;\; t \geq 0. \end{cases} \tag{3.24}$$

$$\begin{cases} \hat{\mathbf{x}}(s) = (s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\hat{\mathbf{u}}(s) + (s\mathbf{E} - \mathbf{A})^{-1}\mathbf{E}\mathbf{x}_0, \\ \hat{\mathbf{y}}(s) = \big(\mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\big)\hat{\mathbf{u}}(s) + \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{E}\mathbf{x}_0, \;\; t \geq 0. \end{cases} \tag{3.25}$$

So, here we can define the following transfer functions. The one comes from the input and is denoted as $\mathbf{H}_u$, and the other comes from the initial conditions and is denoted as

$\mathbf{H}_{x_0}$.

$$\begin{aligned}\mathbf{H}_u(s) &= \mathbf{C}(s\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}+\mathbf{D},\\ \mathbf{H}_{x_0}(s) &= \mathbf{C}(s\mathbf{E}-\mathbf{A})^{-1}\mathbf{E}\mathbf{x}_0.\end{aligned} \tag{3.26}$$

Instead of folding the $\mathbf{D}$ term in the $\mathbf{E}$, we assume it is zero. Further, we can rewrite the remaining output as follows:

$$\mathbf{y}(s) = \mathbf{C}(s\mathbf{E}-\mathbf{A})^{-1}\underbrace{\left[\begin{array}{cc}\mathbf{B} & \mathbf{E}\mathbf{x}_0\end{array}\right]}_{\tilde{\mathbf{B}}}\left[\begin{array}{c}\hat{\mathbf{u}}(s)\\ \mathbf{1}\end{array}\right] \tag{3.27}$$

Remark 3.16 (The initial condition as an input?):
The "position" of the initial condition indicates that $\mathbf{x}_0$ acts as a system's parameter and enters the invariant system's part of the transfer function. ◊

Thus, the total transfer function is: $\mathbf{H}(s) = \mathbf{C}(s\mathbf{E}-\mathbf{A})^{-1}\tilde{\mathbf{B}} \in \mathbb{C}^{1\times(p+1)}$.

The interpolation problem we want to solve is a special MIMO case that has already been addressed. We want to find an interpolant $\mathbf{H}_r$ that satisfies the following interpolation conditions along some directions $\boldsymbol{\ell}_i$, $\mathbf{r}_i$.

$$\begin{aligned}\mathbf{H}(\sigma_i)\boldsymbol{\ell}_i &= \mathbf{H}_r(\sigma_i)\boldsymbol{\ell}_i \text{ for } i = 1,\dots,r,\\ \mathbf{r}_i^T\mathbf{H}(\mu_j) &= \mathbf{r}_i^T\mathbf{H}_r(\mu_j) \text{ for } j = 1,\dots,q.\end{aligned} \tag{3.28}$$

Example 3.3 (A toy MISO example for system and initial conditions identification.):
We consider the linear system with the following matrices:

$$\begin{aligned}&\mathbf{E} = \left[\begin{array}{cc}1 & 0\\ 0 & 1\end{array}\right],\ \mathbf{A} = \left[\begin{array}{cc}-1 & -10\\ 10 & -1\end{array}\right],\ \tilde{\mathbf{B}} = \left[\begin{array}{cc}\mathbf{B} & \mathbf{E}\mathbf{x}_0\end{array}\right] = \left[\begin{array}{cc}1.0 & 2.0\\ 1.0 & 1.0\end{array}\right],\\ &\mathbf{C}^T = \mathbf{B},\ \mathbf{x}_0 = \left[\begin{array}{c}2.0\\ 1.0\end{array}\right].\end{aligned} \tag{3.29}$$

By sampling the "extended" transfer function with the $\tilde{\mathbf{B}}$, we have the following measurements for each. The sampling grid consist of 100 points distributed logarithmically within the $[10^{-2}, 10^{2}]$.

The Loewner matrices were constructed, and the singular value decomposition gave the following decay following Algorithm 3.2. As the Loewner singular value decay drops to machine precision after the second singular value, the underlying McMillan degree of the system is 2. The identified linear model is of order $r = 2$ with matrices:

$$\begin{aligned}&\mathbf{E}_r = \left[\begin{array}{cc}1.93 & -5.59\\ 6.13 & -1.81\end{array}\right],\ \mathbf{E}_r^{-1}\mathbf{A}_r = \left[\begin{array}{cc}-4.33 & 9.4\\ -11.8 & 2.33\end{array}\right],\ \mathbf{E}_r^{-1}\tilde{\mathbf{B}}_r = \left[\begin{array}{cc}-0.263 & -0.525\\ -0.372 & -0.466\end{array}\right],\\ &\mathbf{C}_r^T = \left[\begin{array}{c}-2.54\\ -3.58\end{array}\right],\ \mathbf{x}_{0r} = \left[\begin{array}{c}-0.525\\ -0.466\end{array}\right].\end{aligned} \tag{3.30}$$

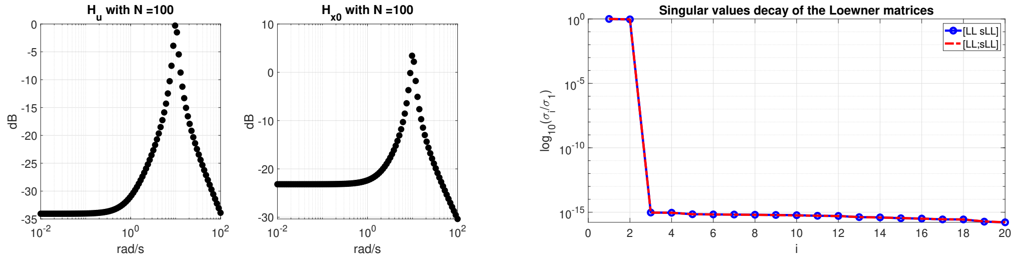


Figure 3.31.: Left: The two transfer functions (MISO) are input to output and initial conditions to output. Right: The Loewner singular value decay captures the minimality.

Remark 3.17 (Recovery of initial conditions?):
It is important here to observe that the initial condition we observe from the original is the same as the model constructed by Loewner. that is, $y_r(0) = \mathbf{C}_r\mathbf{x}_{0r} = 3 = y(0)$. $\Diamond$

Next are some summarizing results.

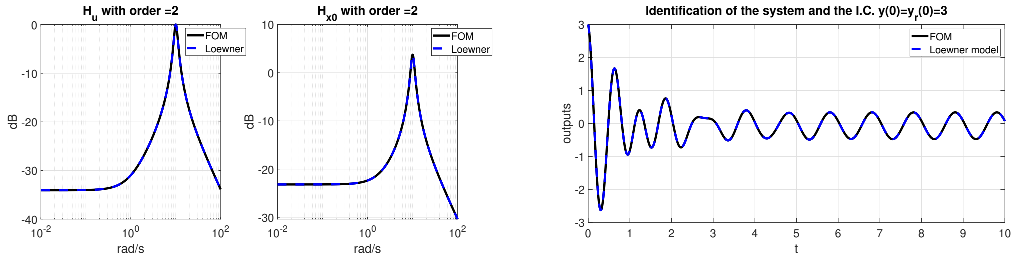


Figure 3.32.: Left: Identification of the MISO TF. Right: Time domain evaluation with the theoretical.

Remark 3.18 (What if we change the initial conditions?):
As the $\mathbf{x}_0$ enters the invariant part, it acts as a parameter, and the transfer function is variant on the initial conditions. Therefore, we can identify a linear system with the same initial condition $y(0) = y_r(0)$. If we change the x_0, it will change the constructed Loewner model. That is explained as the $\mathbf{x}_0$ enters as follows:

$$\mathbf{C}(s\mathbf{E}-\mathbf{A})^{-1}\underbrace{\begin{bmatrix}\mathbf{B} & \mathbf{E}\mathbf{x}_0\end{bmatrix}}_{\breve{\mathbf{B}}}\begin{bmatrix}\hat{\mathbf{u}}(s)\\ \mathbf{1}\end{bmatrix} = \mathbf{C}(s\mathbf{E}-\mathbf{A})^{-1}\underbrace{\begin{bmatrix}\mathbf{B} & \mathbf{E}\end{bmatrix}}_{\breve{\mathbf{B}}}\begin{bmatrix}\hat{\mathbf{u}}(s)\\ \mathbf{x}_0\end{bmatrix} \tag{3.31}$$

The above matrix $\breve{\mathbf{B}}$ enlarges the input dimension by n, but this could be a way to interpret the initial conditions x_0 without affecting the invariants system's properties. The framework works as a full matrix interpolation, not in some tangential directions.$\Diamond$

Therefore, if we employ the above invariant scheme, the new reduced order model from Lowner is:

$$\mathbf{E}_r = \begin{bmatrix} -1.25 & 4.49 \\ -4.64 & 1.53 \end{bmatrix}, \; \mathbf{E}_r^{-1}\mathbf{A}_r = \begin{bmatrix} -4.26 & 9.21 \\ -12.0 & 2.26 \end{bmatrix}, \; \mathbf{E}_r^{-1}\tilde{\mathbf{B}}_r = \begin{bmatrix} -0.382 & -0.327 & -0.055 \\ -0.43 & -0.0558 & -0.374 \end{bmatrix}, \\ \mathbf{C}_r^T = \begin{bmatrix} -2.67 & -2.28 \end{bmatrix}. \tag{3.32}$$

We introduce the following similarity transformation $\mathbf{G}$ to achieve the same coordinates.

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} -0.327 & -0.055 \\ -0.0558 & -0.374 \end{bmatrix}, \\ \mathbf{A}_g &= \mathbf{G}^{-1}\mathbf{A}_i\mathbf{G} = \mathbf{A}, \\ \mathbf{B}_g &= \mathbf{G}^{-1}\mathbf{B}_i = \mathbf{B}, \\ \mathbf{C}_g &= \mathbf{C}_r\mathbf{G} = \mathbf{C}. \end{aligned} \tag{3.33}$$

After we have built the Loewner model, we can change the initial conditions $\mathbf{x}_0$, and the FOM and the ROM after projection will stay close.

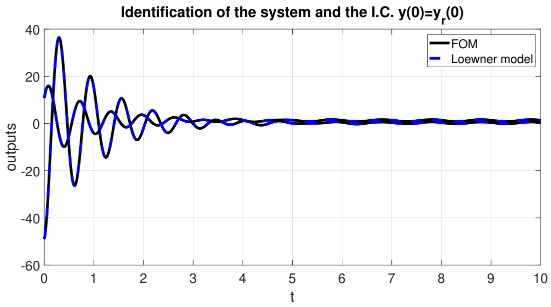


Figure 3.33.: Recovery with known initial conditions.

Remark 3.19 (knowing or not the $\mathbf{x}_0$):
The first approach is recommended for a system with unknown initial conditions where the x_0 acts as a parameter. If the initial conditions are known, the second is recommended where the x_0 plays the role of the input along with the controller. In system theory, inputs should be known. ◊

The following studies [20, 90] are recommended for a more detailed analysis of linear systems with inhomogeneous initial conditions.

CHAPTER 4

IDENTIFICATION AND REDUCTION OF BILINEAR SYSTEMS

Contents

4.1. Introduction

The Loewner framework (LF) Section 3.2, in combination with the Volterra series-(VS) Section 2.3, offers a non-intrusive approximation method capable of identifying bilinear models from time-domain measurements. This method uses harmonic inputs, which establish a natural data acquisition method. For the general class of nonlinear problems with VS representation, the growing exponential approach allows the derivation of the generalized kernels, namely symmetric generalized frequency response functions (GFRFs). In addition, the homogeneity of the Volterra operator determines the accuracy in terms of how many kernels are considered. The weakly nonlinear setup needs only a few kernels to obtain a good approximation. In this direction, the proposed adaptive scheme Section 2.3.2 can improve the estimations of the computationally non-zero kernels. The Fourier transform associates these measurements with the derived GFRFs in system theory, and the LF with linear algebra tools provides state-space dynamical models.

The proposed method aims to extend identification to the case of bilinear systems from time-domain measurements and approximate other general nonlinear systems by using the Carleman linearization scheme Section 2.4.1 that approximates general nonlinear systems with analytical nonlinearities. In the linear case, the LF associates the S-parameters with the linear transfer function by interpolating in the frequency domain. By identifying the linear contribution with the LF, a considerable reduction is achieved using the SVD. The fitted linear system has the same McMillan degree Definition 3.1 as the original linear system. Then, the performance of the linear model is improved by augmenting a particular nonlinear structure. We learn reduced-dimension bilinear models from a potentially large-scale system simulated in the time domain. This is done by fitting a linear model and providing the corresponding bilinear operator afterward. The LF has been extended to infer time-domain data when the underlying system is linear [127].

Extending identification from time domain data in the nonlinear case is challenging. The main difficulties of extending the time-domain Loewner framework [127] in the nonlinear case are; 1st) how to separate the commensurate frequencies in the Fourier spectrum under multi-tone excitation and 2nd) how to enforce interpolation of the symmetric generalized frequency response functions (GFRFs). Next, we present solutions to these problems.

The bilinear Loewner framework [12] addresses bilinear identification and reduction but to artificial data that can be produced by simulating an accessible high-fidelity model intrusively. Thus, in our case with input-output time domain data, this is not the case.

One way to combine the bilinear Loewner framework with the time domain data will be possible if we realize the regular transfer functions that bilinear Loewner needs with the symmetric transfer function that can be measured. We propose a possible way that theoretically is grounded but inherits high numerical complexity.

When state access is possible, we devise a spectral method that can identify the bilinear state equation from the measured frequency domain that constructs models that remain input-independent in contrast to many state access methods; the resulting models are input-dependent.

Continuously in this chapter, we introduce a method that identifies or reduces discrete bilinear systems from the input-output time domain data based on the realization theory from [95]. Given the high complexity of the method, in terms of simulation, we use the power of neural networks in learning input-output maps through the universal approximation theorem in combination with Isidori's realization theory to construct nonlinear state-space bilinear models. The developed method is tested to benchmark examples and compared with state-of-the-art methods such as the subspace identification method.

4.2. Modeling continuous bilinear control systems from input-output data

In recent years, projection-based Krylov methods have extensively been applied for model reduction of bilinear systems. We mention the following contributions [12, 1, 15, 22, 23, 38, 56, 130] and the references within. The set of matrices that describes single input single output (SISO) bilinear systems is $\mathbf{\Sigma}_b = (\mathbf{A}, \mathbf{N}, \mathbf{b}, \mathbf{c}, \mathbf{E})$ and characterized by the following continuous in time system of equations:

$$\mathbf{\Sigma}_b : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)u(t) + \mathbf{b}u(t), \\ \quad y(t) = \mathbf{c}\mathbf{x}(t),\ \mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0},\ t \geq 0. \end{cases} \tag{4.1}$$

where $\mathbf{E}, \mathbf{A}, \mathbf{N} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$, $\mathbf{c} \in \mathbb{R}^{1\times n}$, and $\mathbf{x} \in \mathbb{R}^{n\times 1}, u, y \in \mathbb{R}$. In what follows, we restrict our analysis to systems with non-singular $\mathbf{E}$ matrices (e.g., identity matrix).

4.2.1. The growing exponential approach

The growing exponential approach's properties can be readily adapted to the problem of *finding transfer functions* for constant-parameter (stationary) state equations. Let us consider the bilinear model in Eq. (4.1) with zero initial conditions $\mathbf{x}_0 = \mathbf{0}$.

A single-tone input with amplitude $A < 1$ is considered in Eq. (2.31).

$$u(t) = A\cos(\omega t) = \frac{A}{2}e^{j\omega t} + \frac{A}{2}e^{-j\omega t} = ae^{j\omega t} + ae^{-j\omega t}, \tag{4.2}$$

where $a = A/2$ and $a \in (0, \epsilon)$ with $0 < \epsilon < 1/2$ and for all $t \geq 0$. The steady-state solution for the differential equation in Eq. (4.1) can be written as:

$$\mathbf{x}(t) = \sum_{p,q\in\mathbb{N}}^{\infty} \mathbf{G}_n^{p,q}(\underbrace{j\omega, \ldots, j\omega}_{p-times}, \underbrace{-j\omega, ..., -j\omega}_{q-times})a^{p+q}e^{j\omega(p-q)t} \tag{4.3}$$

The symbol[1] $\mathbf{G}_n^{p,q}$ denotes the n^{th} input to state frequency response containing p-times the frequency ω and q-times the frequency $-\omega$. By substituting in Eq. (4.1) and collecting the terms of the same exponential (as the $e^{j\omega_m t}$), we can derive the input to state frequency responses $\mathbf{G}_n$ for every n as follows:

$$\sum_{p,q\in\mathbb{N}}^{\infty}(j\omega(p-q)\mathbf{E}-\mathbf{A})\,\mathbf{G}_n^{p,q}a^{p+q}e^{j\omega(p-q)t}=\mathbf{b}(ae^{j\omega t}+ae^{-j\omega t})+$$
$$+\mathbf{N}\left(\sum_{p,q\in\mathbb{N}}^{\infty}\mathbf{G}_n^{p,q}a^{p+q+1}e^{j\omega(p+1-q)t}+\sum_{p,q\in\mathbb{N}}^{\infty}\mathbf{G}_n^{p,q}a^{p+q+1}e^{j\omega(p-q-1)t}\right).$$

For the first choices of p and q up to $p+q\leq 2,\ (1,0),(0,1),(2,0),(0,2),(1,1)$ and by denoting the resolvent $\mathbf{\Phi}(j\omega)=(j\omega\mathbf{E}-\mathbf{A})^{-1}\in\mathbb{C}^{n\times n}$, c.t. conjugate terms, we derive the first set of terms:

$$\mathbf{\Phi}(j\omega)^{-1}\mathbf{G}_1^{1,0}ae^{j\omega t}+\mathbf{\Phi}(2j\omega)^{-1}\mathbf{G}_2^{2,0}a^2e^{2j\omega t}+\mathbf{\Phi}(0)^{-1}\mathbf{G}_2^{1,1}a^2+c.t.+\cdots=$$
$$\mathbf{N}\mathbf{G}_1^{1,0}a^2e^{2j\omega t}+\mathbf{N}\mathbf{G}_2^{2,0}a^3e^{3j\omega t}+\mathbf{N}\mathbf{G}_2^{1,1}a^3e^{j\omega t}+c.t.+\cdots+\mathbf{b}ae^{j\omega t}+c.t.$$

Collecting the same powers in both exponential and polynomial magnitudes, we compute the first and the second time/input invariant GFRFs:

$$\begin{aligned}\mathbf{G}_1^{1,0}(j\omega)&=\mathbf{\Phi}(j\omega)\mathbf{b},\\ \mathbf{G}_2^{2,0}(j\omega)&=\mathbf{\Phi}(2j\omega)\mathbf{N}\mathbf{G}_1^{1,0}=\mathbf{\Phi}(2j\omega)\mathbf{N}\mathbf{\Phi}(j\omega)\mathbf{b}.\end{aligned}\tag{4.4}$$

Then, the following input to state transfer functions $\mathbf{G}_n$ using induction are:

$$\begin{aligned}\mathbf{G}_n^{n,0}(j\omega)&=\mathbf{\Phi}(nj\omega)\mathbf{N}\mathbf{\Phi}((n-1)j\omega)\mathbf{N}\cdots\mathbf{N}\mathbf{\Phi}(j\omega)\mathbf{b},\\ \mathbf{G}_n^{0,n}(j\omega)&=\mathbf{\Phi}(-nj\omega)\mathbf{N}\mathbf{\Phi}(-(n-1)j\omega)\mathbf{N}\cdots\mathbf{N}\mathbf{\Phi}(-j\omega)\mathbf{b},\\ \mathbf{G}_n^{p,q}(j\omega)&=\mathbf{\Phi}((p-q)j\omega)\mathbf{N}\left[\mathbf{G}_{n-1}^{p,q-1}(j\omega)+\mathbf{G}_{n-1}^{p-1,q}(j\omega)\right],\ p,q\geq 1,\end{aligned}\tag{4.5}$$

for $n\geq 1$ and $p+q=n$. By multiplying with the output vector $\mathbf{c}$, we can further derive the input-output generalized frequency responses GFRFs as:

$$\begin{aligned}H_n^{n,0}(j\omega)&=\mathbf{c}\mathbf{\Phi}(nj\omega)\mathbf{N}\mathbf{\Phi}((n-1)j\omega)\mathbf{N}\cdots\mathbf{N}\mathbf{\Phi}(j\omega)\mathbf{b},\\ H_n^{0,n}(j\omega)&=\mathbf{c}\mathbf{\Phi}(-nj\omega)\mathbf{N}\mathbf{\Phi}(-(n-1)j\omega)\mathbf{N}\cdots\mathbf{N}\mathbf{\Phi}(-j\omega)\mathbf{b},\\ H_n^{p,q}(j\omega)&=\mathbf{c}\mathbf{\Phi}((p-q)j\omega)\mathbf{N}\left[\mathbf{G}_{n-1}^{p,q-1}(j\omega)+\mathbf{G}_{n-1}^{p-1,q}(j\omega)\right],\ p,q\geq 1.\end{aligned}\tag{4.6}$$

At this point, we can write the Volterra series similarly as in Section 2.3.1 by using the above specific structure of the GFRFs derived with the growing exponential approach for the bilinear case. An important property is that the n^{th} kernel is a multivariate function of order n. Identifying the n^{th}-order FRF involves an n-dimensional frequency space [37].

[1] $\mathbf{G}_n^{p,q}=\mathbf{G}(\underbrace{j\omega,...,j\omega}_{p-times};\underbrace{-j\omega,...,-j\omega}_{q-times})$.

Therefore, we next derive the general 2nd symmetric kernel for the bilinear case with a double-tone input.

$$u(t) = A_1\cos(\omega_1 t) + A_2\cos(\omega_2 t) = \sum_{i=1}^{2} \alpha_i (e^{j\omega_i t} + e^{-j\omega_i t}), \tag{4.7}$$

where $\alpha_1 = \frac{A_1}{2}$ and $\alpha_2 = \frac{A_2}{2}$. In that case, with the growing exponential approach, the state solution in steady state is

$$\mathbf{x}(t) = \sum_{m_1,\dots,m_4 \in \mathbb{N}}^{\infty} \mathbf{G}_n^{m_1,m_2,m_3,m_4} \alpha_1^{m_1+m_2} \alpha_2^{m_3+m_4} e^{j((m_1-m_2)\omega_1+(m_3-m_4)\omega_2)t}. \tag{4.8}$$

We are looking for the input to state frequency response function $\mathbf{G}(j\omega_1, j\omega_2)$. By substituting to the bilinear model in Eq. (4.1) and collecting the appropriate terms while at the same time using the symmetry $\mathbf{G}(j\omega_1, j\omega_2) = \mathbf{G}(j\omega_2, j\omega_1)$, we conclude that:

$$\mathbf{G}_2(j\omega_1, j\omega_2) = \frac{1}{2}\left[(j\omega_1 + j\omega_2)\mathbf{E} - \mathbf{A}\right]^{-1} \mathbf{N} \left[(j\omega_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} + (j\omega_2\mathbf{E} - \mathbf{A})^{-1}\mathbf{b}\right], \tag{4.9}$$

Where by using the resolvent notation and multiplying with $\mathbf{c}$, we derive the *2nd order symmetric generalized frequency response function as:*

$$H_2(j\omega_1, j\omega_2) = \frac{1}{2}\mathbf{c}\mathbf{\Phi}(j\omega_1 + j\omega_2)\mathbf{N}\left[\mathbf{\Phi}(j\omega_1)\mathbf{b} + \mathbf{\Phi}(j\omega_2)\mathbf{b}\right]. \tag{4.10}$$

4.2.2. Identification of the bilinear operator

The difference between linear and bilinear models is the presence of the product between the input and the state that is scaled by the matrix $\mathbf{N}$. As the LF can identify the linear part $(\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{E})$ of the bilinear model, the only thing that remains is identifying the matrix $\mathbf{N}$. The matrix $\mathbf{N}$ enters linearly in the following kernels (as $\mathbf{E}$ has been considered invertible, for simplicity, it is assumed $\mathbf{E} = \mathbf{I}$):

- With a single-tone input, the kernel $H_2^{1,1}$ can be written as:

$$H_2(j\omega_1, -j\omega_1) = \frac{1}{2}\mathbf{c}\left(-\mathbf{A}\right)^{-1}\mathbf{N}\left((j\omega_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + (-j\omega_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\right). \tag{4.11}$$

  and the kernel $H_2^{2,0}$ as:

$$H_2(j\omega_1, j\omega_1) = \mathbf{c}\left(2j\omega_1\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{N}(j\omega_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}. \tag{4.12}$$

- While with a double-tone input, the general kernel H_2 can be written as:

$$H_2(j\omega_1, j\omega_2) = \frac{1}{2}\mathbf{c}\Big((j\omega_1 + j\omega_2)\mathbf{I} - \mathbf{A}\Big)^{-1}\mathbf{N}\Big((j\omega_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + (j\omega_2\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\Big). \tag{4.13}$$

We introduce the following notation:

$$\begin{aligned} \mathcal{O}(j\omega_1, j\omega_2) &= \frac{1}{2}\mathbf{c}\Big(j(\omega_1 + j\omega_2)\mathbf{I} - \mathbf{A}\Big)^{-1} \in \mathbb{C}^{1\times n}, \\ \mathcal{R}(j\omega_1, j\omega_2) &= \Big((j\omega_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + (j\omega_2\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\Big) \in \mathbb{C}^{n\times 1}. \end{aligned} \tag{4.14}$$

Then, Eq. (4.13) can be compactly rewritten as:

$$H_2(j\omega_1, j\omega_2) = \mathcal{O}(j\omega_1, j\omega_2)\mathbf{N}\mathcal{R}(j\omega_1, j\omega_2). \tag{4.15}$$

Assume that k measurements of the function H_2 are available (measured) for k different pairs (ω_1, ω_2). By vectorizing with respect to the measurement set, we have for the k^{th} measurement:

$$\underbrace{H_2(j\omega_1^{(k)}, j\omega_2^{(k)})}_{\mathbf{Y}^{(k)}} = \underbrace{\mathcal{O}(j\omega_1^{(k)}, j\omega_2^{(k)})}_{\mathcal{O}^{(k)}_{1,n}} \underbrace{\mathbf{N}}_{n\times n} \underbrace{\mathcal{R}(j\omega_1^{(k)}, j\omega_2^{(k)})}_{\mathcal{R}^{(k)}_{n,1}},$$

$$\text{For all } k \text{ measurements} \rightarrow \mathbf{Y}_{(1:k,1)} = \underbrace{\Big(\mathcal{O}^{(k)}_{(1,n)} \otimes \mathcal{R}^{T(k)}_{(1,n)}\Big)}_{(1:k,n^2)} \underbrace{vec\,(\mathbf{N})}_{(1:n^2,1)}{}^{2} \tag{4.16}$$

Note that equations (4.11), (4.12), (4.13) can be equivalently rewritten as the one linear matrix equation given in Eq. (4.16). By filling out the above matrix $\left[\mathcal{O} \otimes \mathcal{R}^T\right]$ with the information from $H_2(j\omega_1, -j\omega_1)$ and from $H_2(j\omega_1, j\omega_1)$ as well, the solution can be improved. Hence, we can solve Eq. (4.16) with full rank and identify the matrix $\mathbf{N}$. All the symmetry properties of the kernels are appropriately used, e.g., conjugate-real symmetry. For n denoting the dimension of the bilinear model and k the number of measurements, we have the following two cases:

1. $k < n^2$ under-determined → least squares (LS) solution (minimizing the 2-norm) as in [101],
2. $k \geq n^2$ (over)determined - rank completion → identification of $\mathbf{N}$,

Theorem 4.1 (Bilinear identification):
Let $\Sigma_b = (\mathbf{A}, \mathbf{N}, \mathbf{b}, \mathbf{c}, \mathbf{E})$ be a bilinear system of dimension n for which the linear subsystem $\Sigma_l = (\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{E})$ is *fully controllable and observable.* Then, for $k \geq n^2$ measurements so that $(j\omega_1^{(k)}, j\omega_2^{(k)})$ are distinct complex pairs with $(\omega_1^{(k)}, \omega_2^{(k)}) \in \mathbb{R}^2_+$ and $\omega_1^{(k)} \neq \omega_2^{(k)}$, the following holds:

$$rank\left(\underbrace{\begin{bmatrix} \mathcal{O}^{(1)} \otimes \mathcal{R}^{T(1)} \\ \mathcal{O}^{(2)} \otimes \mathcal{R}^{T(2)} \\ \vdots \\ \mathcal{O}^{(k)} \otimes \mathcal{R}^{T(k)} \end{bmatrix}}_{(1:k\geq n^2, n^2)}\right) = n^2. \tag{4.17}$$

◊

[2] The vectorization is row-wise, $vec(\mathbf{N}) = \begin{bmatrix} \mathbf{N}(1, 1:n) & \cdots & \mathbf{N}(n, 1:n) \end{bmatrix}^T \in \mathbb{R}^{n^2\times 1}$.

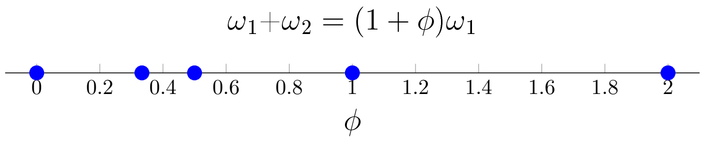


Figure 4.1.: This figure shows the constraints of ϕ (e.g., $\phi = 0, 1/3, 1/2, 1, 2, 3, \ldots$, etc.). By choosing ϕ's within the blue dots, we construct frequency bandwidths with a unique $(\omega_1 + \omega_2)$ up to the specified mixing order.

Theorem 4.1 can be proved in a similar way as in [42]. As the above result indicates, one would need at least n^2 measurements to identify the matrix $\mathbf{N}$ corresponding to the bilinear system of dimension n.

4.2.3. A separation strategy for the 2nd kernel

To identify the n^{th} Volterra kernel, we need an n-tone input signal. To identify the 2nd kernel, the input signal must be chosen at least as a double tone Eq. (4.7). The propagating harmonics are: $e^{(j(m_1-m_2)\omega_1+j(m_3-m_4)\omega_2)t}$ or more compactly $e^{(\pm kj\omega_1 \pm lj\omega_2)t}$, where $k, l \in \mathbb{N}$. The aim is to differentiate the $(\omega_1 + \omega_2)$ harmonic from the other harmonics. More precisely, we want the following result to hold:

$$\omega_1 + \omega_2 \neq k\omega_1 + l\omega_2, \ \forall (k, l) \in \mathbb{Z} \times \mathbb{Z} \setminus \{1, 1\}. \tag{4.18}$$

Suppose $\omega_2 = \phi\omega_1, \ \phi \in \mathbb{R}$. The suitable ϕ's where Eq. (4.18) holds are:

$$\omega_1 + \phi\omega_1 = k\omega_1 + l\phi\omega_1 \Rightarrow 1 + \phi = k + l\phi \Rightarrow \phi = \frac{k-1}{1-l}, \ k, l \in \mathbb{Z} \setminus \{1\}. \tag{4.19}$$

By choosing ϕ so that the equality in Eq. (4.19) doesn't hold, with harmonic mixing index $m = k + l$, it makes the harmonic $(\omega_1 + \omega_2)$ uniquely defined in the frequency spectrum up to the m^{th} kernel.

To visualize this feature, we choose $\omega_1 = 1$ and $\omega_2 = \omega_1\phi = \phi$ for harmonic mixing index $m = 4$. Then, the constraints of ϕ are depicted in Fig. 4.1 with blue dots. Next, in Fig. 4.2 and on the left pane, one ϕ constraint that occurs commensurate harmonics is depicted with the 2nd and the 3rd kernel to contribute at the same harmonic. On the right pane, the harmonic is uniquely defined at $(\omega_1 + \omega_2)$ from the 2nd kernel up to the mixing order $m = 4$.

The next result allows us to construct sweeping frequency schemes to get enough measurements for the $H_2(j\omega_1, j\omega_2)$. So, for every $\omega_1 > 0$ the following should hold:

$$\omega_2 \in (\phi_{i-1}\omega_1, \phi_i\omega_1), \ i = 1, \ldots \tag{4.20}$$

Where ϕ_i are the constraints (see Fig. 4.1 blue dots).

Remark 4.2 (Kernel separation):
In the proposed framework, the separation of the kernels that contribute at $(\omega_1 + \omega_2)$ harmonic is forced only under a specific mixing order m. We do not offer any general

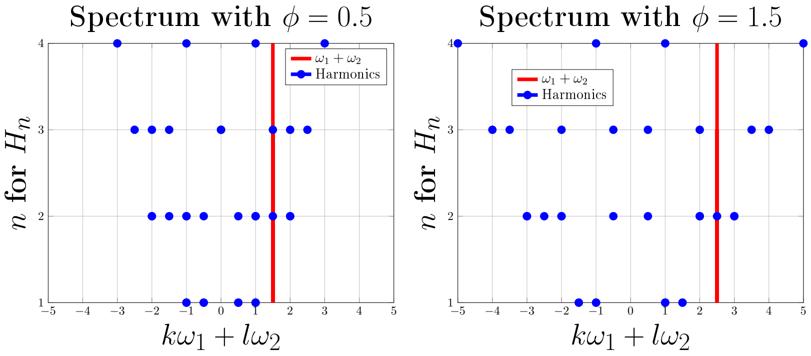


Figure 4.2.: Left pane: Overlapping kernels contributing to the same harmonic with invalid $\phi = 0.5$. Right pane: Uniquely defined harmonic at $(\omega_1 + \omega_2)$ with valid $\phi = 1.5$. Here, it holds $(n = k + l)$.

solution to this separation problem for multi-tone input, although techniques have been introduced, such as in [37]. There, it was also stated that the solution of the complete separation of harmonics is generally not possible. $\Diamond$

4.2.4. The Loewner-Volterra algorithm for bilinear modeling

We start with a set of single-tone inputs $u(t) = \alpha_\ell \cos(\omega_1^{(i)} t)$, $i = 1, ..., k$, with $\alpha_\ell < 1$. For those k measurements, we can estimate the linear kernel $H_1(j\omega_1^{(i)})$, the $H_2(j\omega_1^{(k)}, j\omega_1^{(k)})$, and the $H_2(j\omega_1^{(k)}, -j\omega_1^{(k)})$ by simply measuring the first harmonic as $\mathbf{Y}_1$, the second harmonic as $\mathbf{Y}_2$, and the DC term as $\mathbf{Y}_0$, from the frequency spectrum as in Fig. 2.1. To improve the accuracy of the estimations for the kernels above, we could upgrade to an ℓ-stage approximation by varying the amplitude α_ℓ as explained in Section 2.3.2. This approach is necessary whenever higher harmonics are considered numerically non-zero, hence meaningful. The reason for this is that the first harmonic is hence corrupted by noise introduced by the term $H_3^{2,1}$ and the rest of the terms which appear on the second row of the matrix $\mathbf{P}_\omega$ in Eq. (2.42).

Since the LF reveals the underlying order of the linear system denoted with r, the value of k should be at least equal to $2r$. Then, we can decide on the order r of the reduced system by analyzing the singular value decay. Up to the previous step, we have identified the linear part with the LF, and we have filled the LS problem Eq. (4.16) with measurements from the diagonal of the second kernel and the perpendicular to the diagonal axis $(\omega_1, -\omega_1)$. Those measurements contribute to the problem but with an under-determined (rank deficient) least-squares problem. We need more measurements from H_2 to complete the rank of (r^2) degrees of freedom. Thus, the unique solution will lead to the identification of $\mathbf{N}$. So, we proceed by measuring the H_2 off the diagonal $(\omega_1 \neq \omega_2)$ with a double-tone input as $u(t) = \alpha_\ell \cos(\omega_1^{(k)} t) + \beta_\ell \cos(\omega_2^{(k)} t)$, for a set of frequency pairs (ω_1, ω_2) $(\# \ r^2)$.

We follow the solution proposed in section Section 4.2.3 (up to a mixing degree) to deal with this problem. Finally, to get the bilinear operator $\mathbf{N}$, we solve the real[3] completed rank least-squares (LS) problem as described in Eq. (4.16) by using all the symmetric properties of these kernels (i.e., real symmetry, conjugate symmetry, and the fact that $H_2(j\omega_1, j\omega_2) = H_2(j\omega_2, j\omega_1)$). To numerically improve the ill-conditioned matrices involved, the inversion can be under a threshold in a classical regularization manner (e.g., Thikhovov or thresholding SVD as the L-curve regularization). An algorithm that summarizes the above procedure is presented below.

The computational effort of the proposed method This section analyzes the proposed method's computational effort by analyzing each step. We comment on the applicability of large-scale problems and their relation with real-world scenarios.

Simulation of processes with harmonic inputs constitutes a classical technique in many engineering applications; data acquisition in the time domain is a standard procedure. Nevertheless, using advanced electronic devices such as vector network analyzers (VNAs), frequency-domain data can also be obtained (directly). The Loewner framework applied in the case where frequency-domain data obtained from VNAs offers an excellent identification and reduction tool in the linear case (with many applications in electrical, mechanical, or civil engineering). In the context of the current paper, we deal with time-domain data for a particular class of nonlinear problems.

Data collection is the most expensive procedure to identify and reduce bilinear systems from time-domain measurements. This is done by simulating time-domain models with Euler's method (bilinear models such as the ones approximating Burgers' equation). Nevertheless, the high computational cost of simulating large dimensional systems in the time domain could be alleviated using parallel processing (e.g., for multiple computational clusters). Estimating transfer function values by computing the Fourier transform remains robust. In addition, the LF can adaptively detect the decay of the singular values; hence, the procedure can be terminated for a specifically reduced order $r \ll n$.

Initially, a linear system of reduced dimension r is fitted using the LF. For the rest of the proposed algorithm, note that we will use the lower dimension r to our advantage; hence, the method remains robust. The next step is to compute the matrix $\mathbf{N}$ that characterizes the nonlinearity of bilinear systems. As the fitted linear system is dimension r, we must detect exactly r^2 unknowns (matrix entries $\mathbf{N}$).As presented in section 4.3, this boils down to solving a full-rank LS problem that can be easily dealt with.

The newly proposed method aims to train bilinear models from time-domain data accurately. We offer a first-step approach toward identifying such systems within the Volterra series approximation approach. Large-scale systems are often sparse (due to spatial domain semi-discretization); hence, reduction techniques can be applied. The new method deals with the inherent redundancies through the linear subsystem (compression using SVD). Afterward, it updates the nonlinear behavior by introducing an appropriate low-dimensional bilinear matrix that improves the overall approximation. Note also that the new method relies on the *controllability/observability* of the fitted linear system. Additionally, noise values up to a particular threshold can be handled as presented in

[3] Enforcing real-valued models have been discussed in [13, 102]; here, we follow the same approach.

Algorithm 4.1: Bilinear modeling from time-domain data.

```
Input: Use as control input the signals:
       $u(t) = \alpha_\ell \cos(\omega_1^{(k)} t) + \beta_\ell \cos(\omega_2^{(k)} t),\ t \geq 0$, by sweeping the small
       amplitudes ($< 1$) and a particular range of frequencies.
Output: A reduced bilinear system of dimension-$r$: $\Sigma_{b_r} : (\mathbf{A}_r, \mathbf{N}_r, \mathbf{b}_r, \mathbf{c}_r, \mathbf{E}_r)$
1: Apply one-tone input $u(t)$ with $\beta_\ell = 0$, $\omega_1^{(k)}$ for $k = 1, \ldots, n$, and collect the
   snapshots $y(t)$ in steady state.
2: Apply Fourier transform and collect the following measurements:
       • DC term: $Y_O(0 \cdot j\omega_1^{(k)})$,
       • 1st harmonic: $Y_I(1 \cdot j\omega_1^{(k)})$,
       • 2nd harmonic: $Y_{II}(2 \cdot j\omega_1^{(k)})$,
         ⋮
       • $m^{\text{th}}$ harmonic: $Y_{m^{th}}(m \cdot j\omega_1^{(k)})$ (last numerically non-zero harmonic).
3: If the 2nd harmonic or higher harmonics are nonzero, the system is nonlinear. By
   sweeping the amplitude and using the adaptive scheme (stage $\ell$-approximation see
   Section 2.3.2) in Eq. (2.43), the estimations of the first and the second kernel can
   be improved. If the 2nd and higher harmonics are equal to zero, the bilinear matrix
   $\mathbf{N}$ remains zero, and the underlying system is linear.
4: Apply the linear LF, see Algorithm 3.1 by using the measurements (e.g.,
   $H_1(j\omega_1^{(k)}) \approx 2Y_I(j\omega_1^{(k)})/\alpha_\ell$ for the 2nd stage approximation $Y_m \approx 0$ for $m > 2$) and
   get the order $r$ linear model.
5: If the system is nonlinear, fitting a bilinear matrix $\mathbf{N}$ will improve the accuracy.
   Apply the 2-tone input $u(t) = \alpha_\ell \cos(\omega_1^{(k)} t) + \beta_\ell \cos(\omega_2^{(k)} t)$ to get enough
   measurements ($\leq r^2$) to produce a full-rank LS problem. Measure the $(\omega_1 + \omega_2)$
   harmonic as explained in section 4.2.3 and get the estimations for the 2nd kernel as:
   $H_2(j\omega_1^{(k)}, j\omega_2^{(k)}) \approx 2Y_{II}(j\omega_1^{(k)}, j\omega_2^{(k)})/(\alpha_\ell \beta_\ell)$.
6: Solve the full rank least squares problem as described in Eq. (4.16) and compute
   the real-valued bilinear matrix $\mathbf{N}$. When the inversion is not exact due to numerical
   issues, the least squares solution is obtained with a thresholding SVD.
```

section 5; further analysis on noise-related issues is left for future research.

Convergence of Volterra series representation and BIBO stability of bilinear systems
A classical theorem for bilinear systems is the following.

Theorem 4.3 (BIBO stability for bilinear systems):
As shown in [144], the bilinear system Eq. (4.28) is bounded-input/bounded-output (BIBO) stable, provided that

$$\|\mathbf{N}\|_2 \leq \frac{\alpha}{C\beta}, \tag{4.21}$$

where $\mathbf{A}$ is Hurwitz stable (i.e., has eigenvalues in the left half of the complex plane) and in addition, $\|e^{\mathbf{A}t}\|_2 \leq \beta e^{-\alpha t},\ t > 0$, and $\|u\|_2 \leq C$. Thus, any bilinear system with stable $\mathbf{A}$ is BIBO stable for small enough control inputs. ◊

4.2.5. Numerical examples

4.2.5.1. Identifying a low-order bilinear system

Example 4.1 (Identifying a low-order bilinear toy example):
This experiment aims to identify a simple bilinear model from time-domain measurements. Consider the following controllable/observable bilinear model Eq. (4.1) of dimension-2 with a *non-symmetric* matrix $\mathbf{N}$, zero initial condition and matrices as:

$$\mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ \mathbf{A} = \begin{bmatrix} -1 & -10 \\ 10 & -1 \end{bmatrix},\ \mathbf{N} = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ \mathbf{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}. \tag{4.22}$$

We simulate the system in the time domain with input as $u(t) = A\cos(\omega t)$, magnitude $A = 0.01$, frequency $\omega \in \begin{bmatrix} 0.5 & 1 & 1.5 & 2 \end{bmatrix} 2\pi$, and time step $dt = 1e{-}4$. Next, the 2nd-stage approximation results for the linear kernel $\tilde{H}_1$ in comparison with the theoretical values of H_1 are presented in Table 4.1.

Table 4.1.: Measurements of H_1

Frequency ω	$\tilde{H}_1(j\omega)$-2nd stage	$H_1(j\omega)$-theoretical
$0.5 \cdot 2\pi$	$+0.026606 + 0.067106$i	$+0.026574 + 0.067115$i
$1.0 \cdot 2\pi$	$+0.071503 + 0.189600$i	$+0.071258 + 0.189700$i
$1.5 \cdot 2\pi$	$+0.752720 + 0.377300$i	$+0.754030 + 0.380870$i
$2.0 \cdot 2\pi$	$+0.134070 - 0.381970$i	$+0.133780 - 0.382520$i

[a] With 2nd-stage approximation $\tilde{H}_1(j\omega) \approx 2Y_1(j\omega)/A$.

With the estimations of the linear transfer function and by using the LF as the data-driven identification and reduction tool for linear systems, we identify the linear system $(\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}, \tilde{\mathbf{E}})$. We stopped at the 4th measurement because the underlying system is second-order (McMillan degree 2). Otherwise, more measurements will be needed for a sufficient decay of the singular values as in Fig. 4.3. The singular values decay offers

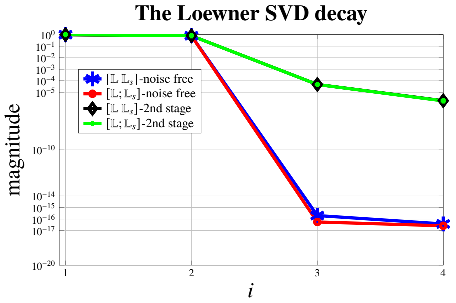


Figure 4.3.: The singular value decay of the LF is a fundamental characterization of the McMillan degree of the underlying linear system. Here, a truncation scheme of order $r = 2$ is recommended where the 2nd stage approximation gave $\sigma_3/\sigma_1 = 4.721 \cdot 10^{-5}$, while for the noise-free case, the third singular values have reached the machine precision.

a choice of reduction. The singular values with magnitude below that threshold are neglected as long as the system simulation is done, with a time step $dt = 1e - 4$.

Construction of the linear system with order $r = 2$, by using the theoretical noise-free measurements (subscript "t") appears next:

$$\tilde{\mathbf{A}}_t = \begin{bmatrix} -1.4513 & -8.8181 \\ 11.363 & -0.54868 \end{bmatrix},\ \tilde{\mathbf{B}}_t = \begin{bmatrix} -0.92979 \\ 1.3967 \end{bmatrix},\ \tilde{\mathbf{C}}_t = \begin{bmatrix} -0.76857 & 0.9203 \end{bmatrix}, \tag{4.23}$$

While by using the measured data with 2nd-stage approximation results to the following:

$$\tilde{\mathbf{A}} = \begin{bmatrix} -1.458 & -8.8137 \\ 11.367 & -0.55162 \end{bmatrix},\ \tilde{\mathbf{B}} = \begin{bmatrix} -0.9342 \\ 1.4 \end{bmatrix},\ \tilde{\mathbf{C}} = \begin{bmatrix} -0.7675 & 0.91611 \end{bmatrix}. \tag{4.24}$$

Remark 4.4 (Identified linear dynamics):
Even if the coordinate system is different, one crucial qualitative result is to compute the poles and zeros of the linear transfer function. For the identified system with the theoretical measurements (noise free), the poles and zeros are precise as the original: $\tilde{p}_t = -1 \pm 10\mathrm{i}$ and the zero are $\tilde{z}_t = -1$ while for the 2nd stage approximation to the linear system, the corresponding results are: $\tilde{p} = -1.0048 \pm 9.9989\mathrm{i}$, $\tilde{z} = -1.0042$. ◊

At this point, we have recovered the linear part of the bilinear system up to accuracy due to the truncation of the Volterra series. The inexact simulations of the continuous system are done with a finite time step $dt = 1e-4$. The Fourier accuracy led to entirely accurate results with a perturbation of the order $\sim O(1e - 3)$ by comparing the theoretical poles and zeros. We proceed by collecting the measurements of the 2nd kernel. Table 4.2 contains measurements of the 2nd kernel with 1-tone input.

We can get $\mathbf{N}$ by solving the least squares problem by minimizing the 2-norm as in [101]. This result was not towards identifying the matrix $\mathbf{N}$; here is the new approach working towards identifying bilinear systems.

Table 4.2.: Measurements of the H_2 on and perpendicular to the diagonal.

Freq. ω	$\tilde{H}_2(j\omega, j\omega)$	$H_2(j\omega, j\omega)$	$\tilde{H}_2(j\omega, -j\omega)$	$H_2(j\omega, -j\omega)$
$0.5 \cdot 2\pi$	$+0.026440 - 0.124490i$	$+0.026570 - 0.124440i$	$+0.032190$	$+0.032177$
$1.0 \cdot 2\pi$	$-0.184590 + 0.298430i$	$-0.184510 + 0.298910i$	$+0.045648$	$+0.045641$
$1.5 \cdot 2\pi$	$+0.178080 + 0.305840i$	$+0.178160 + 0.307170i$	$+0.063936$	$+0.064350$
$2.0 \cdot 2\pi$	$+0.062642 - 0.054219i$	$+0.062588 - 0.054423i$	-0.044927	-0.044998

[b] The estimation of the 2nd kernel is given as: $\tilde{H}_2(j\omega, j\omega) \approx 4Y_2(j\omega, j\omega)/A^2$, on the diagonal, and $\tilde{H}_2(j\omega, -j\omega) \approx 2Y_2(j\omega, -j\omega)/A^2$, which is the DC term.

Remark 4.5 (Can we identify the matrix N?):
The improvement relies on the rank deficiency problem produced by getting the least squares solution without taking under consideration measurements off the diagonal of the 2nd kernel H_2. By filling in the least squares problem in Eq. (4.16) with these different equations, as proposition Theorem 4.1 indicates, the problem solution upgrades to a complete rank inversion, and the answer is affirmative. $\diamondsuit$

Back to our introductory example, the rank of the least squares problem is less than $r^2 = 4$. So, we need to increase the rank. We take measurements (≤ 4) out of the diagonal from the 2nd kernel by using the input $u(t) = A_1 \cos(\omega_1) + B_1 \cos(\omega_2)$. Tab. 4.3 includes the theoretical and measured results.

Table 4.3.: Measurements of the H_2 off the diagonal

Frequencies (ω_1, ω_2)	$\tilde{H}_2(j\omega_1, j\omega_2)$	$H_2(j\omega_1, j\omega_2)$
$(0.2 \cdot 2\pi, 0.3 \cdot 2\pi)$	$+0.030440 - 0.039259i$	$+0.030429 - 0.039237i$
$(0.2 \cdot 2\pi, 0.6 \cdot 2\pi)$	$+0.031002 - 0.080364i$	$+0.031037 - 0.080315i$
$(0.4 \cdot 2\pi, 0.3 \cdot 2\pi)$	$+0.030948 - 0.062869i$	$+0.030961 - 0.062835i$
$(0.4 \cdot 2\pi, 0.6 \cdot 2\pi)$	$+0.026417 - 0.125320i$	$+0.026554 - 0.125260i$

[c] The estimation of the 2nd kernel as $\tilde{H}_2(j\omega_1, j\omega_2) \approx 2Y_2(j\omega_1, j\omega_2)/(A_1 B_1)$. Here we use $\phi = 1.5$ to avoid the harmonic overlapping as explained in section 4.2.3 and amplitudes as $A_1 = B_1 = 0.01$.

The full rank least squares solution gave for the hypothetical noise-free case and for the 2nd stage approximation the following results respectively:

$$\tilde{\mathbf{N}}_t = \begin{bmatrix} -4.1542 & -2.0998 \\ 3.236 & 1.1542 \end{bmatrix}, \; \tilde{\mathbf{N}} = \begin{bmatrix} -4.1557 & -2.1084 \\ 3.2284 & 1.1513 \end{bmatrix} \tag{4.25}$$

Remark 4.6 (Coordinate transformation):
By transforming all the matrices to the same coordinate system as in [97], we conclude with the following:

- **Noise-free case - exact identification**

$$\breve{\mathbf{A}}_t = \begin{bmatrix} -1.0 & -10.0 \\ 10.0 & -1.0 \end{bmatrix},\ \breve{\mathbf{N}}_t = \begin{bmatrix} 1.0 & -2.0 \\ 3.0 & -4.0 \end{bmatrix},\ \breve{\mathbf{B}}_t = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix},\ \breve{\mathbf{C}}_t = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}^T. \tag{4.26}$$

- **Simulated case - approximated identification**

$$\breve{\mathbf{A}} = \begin{bmatrix} -1.0037 & -9.9941 \\ 10.004 & -1.0059 \end{bmatrix},\ \breve{\mathbf{N}} = \begin{bmatrix} 0.99525 & -1.997 \\ 3.006 & -3.9997 \end{bmatrix},\ \breve{\mathbf{B}} = \begin{bmatrix} 0.99925 \\ 1.0003 \end{bmatrix},\ \breve{\mathbf{C}} = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}^T. \tag{4.27}$$

$\Diamond$

Next, in Fig. 4.4, evaluation results for the linear and the second-order generalized transfer function are presented:

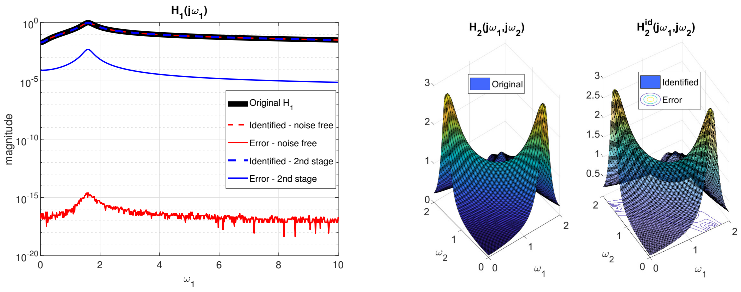


Figure 4.4.: The identified 1st and 2nd kernels with 2nd-stage approximation in comparison with the theoretical kernels.

Finally, time-domain simulations for each system performed in Fig. 4.5 with a larger amplitude than the probing one.

4.2.5.2. Approximation of the viscous Burgers' model equation

Example 4.2 (The viscous Burgers' Equation):
This example illustrates the bilinear modeling and reduction concepts proposed in [12] for the viscous Burgers' equation from time-domain simulations. We simulate the system with 40 measurements as $\omega_k = j2\pi[0.1, 0.2, \ldots, 4]$. We present the corresponding results with initial system dimension $n = 420$ reduced by the proposed method to order $r = 2$ with the first normalized neglected singular value to be $\sigma_3/\sigma_1 = 4.6255 \cdot 10^{-4}$. As the order was chosen $r = 2$, the reduced bilinear matrix $\tilde{\mathbf{N}}$ was introduced by using the following measurements as $\omega_1 = j2\pi[0.2, 0.4]$ and $\omega_2 = j2\pi[0.3, 0.6]$. Evaluation results are presented in Fig. 4.6. Lastly, in Fig. 4.7, a time-domain simulation reveals that the proposed method can improve the accuracy by fitting a nonlinear model. Table 4 contains approximation results both in the frequency and time domains. For the example presented (dimension reduction from $n = 420$ to $r = 2$, i.e., $\sim 99\%$), we offer a comparison of the newly-proposed method (Time-LoewBil) with another way, i.e., the frequency-domain bilinear Loewner framework introduced in [12] (Freq-LoewBil). The

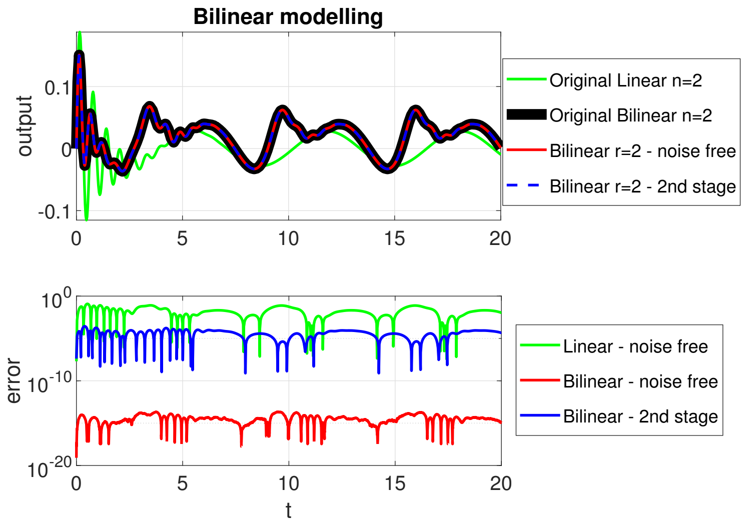


Figure 4.5.: The evaluation of the models with order $r = 2$ performed with input as $u(t) = \cos(t),\ t \in [0, 20]$. The noise-free case has reached machine precision.

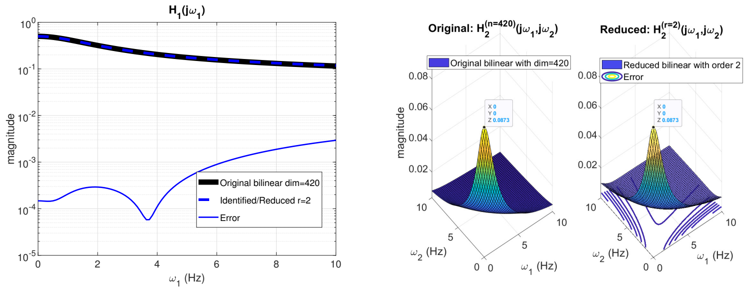


Figure 4.6.: The 1st and the 2nd kernel evaluations compared to the originals.

standard frequency grid was selected as described above, while the sampling values of the transfer functions (in the frequency domain) were corrupted with white noise. The noise magnitude of the latter was selected to match the noise values introduced by performing time-domain simulations with a time step of $dt = 1e - 4$.

4.3. The bilinear time domain Loewner framework

In Chapter 2 and Section 2.3, we defined the Volterra series and their properties for general nonlinear systems. We continue our analysis with bilinear systems with state dimension n. Bilinear systems can be derived directly from a physical system (e.g., inflow-outflow engineering processes) or after approximating a nonlinear analytical model with the Carleman bilinearization scheme.

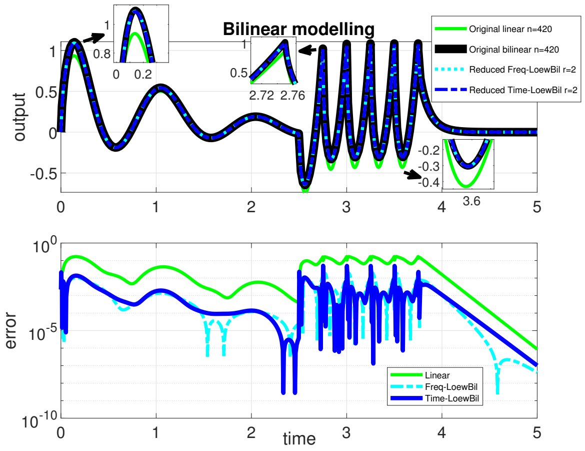


Figure 4.7.: Time-domain simulation for the Burgers' equation example; viscosity parameter ν is set as 1, and the dimension of the semi-discretized model is chosen to be 420. A comparison of the identified/reduced bilinear of order $r = 2$ with the linear and frequency-domain Loewner bilinear is depicted. The input is chosen as: $u(t) = (1 + 2\cos(2\pi t))e^{-t}, t \in [0, 2.5],\ u(t) = 4\text{sawtooth}(8\pi t), t \in [2.5, 3.75],\ u(t) = 0, t \in [3.75, 5]$.

Remark 4.7 (Carleman bilinearization and linear minimality):
Carleman linearization of autonomous systems or bilinearization of control systems approximates the state polynomial nonlinearities (analytical) with a Taylor expansion approximation as explained in Section 2.4.1. The approximation comes from that higher-state polynomial terms (cubic and higher) are neglected. The resulting system's larger dimension scales exponentially with the original one. Most importantly, the minimality of the linear sub-model does not always explain the bilinear system's minimality. Therefore, approximation of nonlinear systems with methods that rely on the linear minimality degree as in Algorithm 4.1 could not always reveal the theoretical minimal degree of the bilinear system that approximates the underlying nonlinear system. We notice this phenomenon of miss-matching minimalities, especially when lifting strategies (e.g., Carleman or Kerner methods) lead to structured nonlinear embedding of higher-order nonlinearities that forced to preserve a low-order state nonlinearity as quadratic see Chapter 5 or bilinear in the current chapter. ◊

Remark 4.8 (Different types of Volterra kernels):
Using the regular Volterra kernels, the Loewner framework has been extended to the class of bilinear systems in [12]. The type that can be measured from the time domain simulation is symmetric. These different types of Volterra kernels are connected. Thus, we aim to identify the regular kernels (bilinear Loewner) from the symmetric (measured kernels) type and apply the nonlinear Loewner framework. ◊

To keep the mathematical complexity low, we consider the single-input-single-output (SISO) case, and in the latter, the explicit derivation of the corresponding kernels follows.

The (SISO) bilinear system is

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)u(t) + \mathbf{b}u(t), \\ y(t) = \mathbf{c}\mathbf{x}(t), \ \mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0}, \end{cases} \tag{4.28}$$

where $\mathbf{A}, \ \mathbf{N} \in \mathbb{R}^{n\times n}, \ \mathbf{b}, \ \mathbf{c}^T \in \mathbb{R}^{n\times 1}$. The bilinear system in Eq. (4.28) has state dimension n, the input is $u(t)$, the output is $y(t)$, and starts at $t_0 = 0$ with zero-state initial conditions $\mathbf{x}_0 = \mathbf{0}$.

4.3.1. Regular kernels for the bilinear system

There are several ways to derive Volterra kernels from yielding a Volterra series. Following the procedure and the notation in [8], someone can derive specific types based on the variational approach (Picard iterations) described in [137].

Definition 4.9 (The mth triangular bilinear Volterra kernel):

$$h_m^{\texttt{tri}}(t, \tau_1, \tau_2, \ldots, \tau_m) = \mathbf{c}e^{\mathbf{A}(t-\tau_1)}\mathbf{N}e^{\mathbf{A}(\tau_1-\tau_2)} \cdots \mathbf{N}e^{\mathbf{A}(\tau_{m-1}-\tau_m)}\mathbf{b}, \ m \geq 1. \tag{4.29}$$

$\Diamond$

Using the change of variable $t_{m-i} = \tau_i - \tau_{i+1}$ for $i = 0, 1, \ldots, m-1$, with the convention that $t = \tau_0$, we obtain the so-called regular Volterra kernel as

Definition 4.10 (The mth regular bilinear Volterra kernel):

$$h_m^{\texttt{reg}}(t_1, t_2, \ldots, t_m) = \mathbf{c}e^{\mathbf{A}t_m}\mathbf{N}e^{\mathbf{A}t_{m-1}} \cdots \mathbf{N}e^{\mathbf{A}t_1}\mathbf{b}, \ m \geq 1. \tag{4.30}$$

$\Diamond$

With the identity matrix $\mathbf{I} \in \mathbb{R}^{n\times n}$, we denote the resolvent $\mathbf{\Phi}(s) = (s\mathbf{I} - \mathbf{A})^{-1} \in \mathbb{C}^{n\times n}$, and with the multidimensional Laplace transform in Eq. (4.30), we obtain the mth regular generalized frequency response (mth subsystem regular transfer function) as

$$H_m^{\texttt{reg}}(s_1, s_2, \ldots, s_m) = \mathbf{c}\mathbf{\Phi}(s_m)\mathbf{N}\mathbf{\Phi}(s_{m-1})\mathbf{N} \cdots \mathbf{N}\mathbf{\Phi}(s_2)\mathbf{N}\mathbf{\Phi}(s_1)\mathbf{b}. \tag{4.31}$$

The regular type of kernels yields a Volterra series, and interpolatory model reduction [8] such as the one based on the Loewner matrices can construct reduced models after enforcing interpolation to these multi-variable rational transfer functions Eq. (4.31) as in [12]. The restriction with the above type of kernels starts when a physical measurement environment is concerned. In the original extension of the Loewner framework to bilinear systems [12], data were considered as sample values of the regular kernels over an arbitrary frequency domain $s_1, \ldots, s_m$ from an accessible high fidelity model. Consequently, to have such data, one needs the original operators of the large-scale bilinear system to construct reduced models. The bilinear Loewner framework belongs to the class of intrusive methods. It uses data from direct numerical simulations (DNS).

4.3.2. The bilinear Loewner framework

The framework presented here first appears in [12]. We assume that $k = 2\tilde{k}$ left and right interpolation points are available, which are grouped as follows:

$$\begin{aligned} \text{left points}: \ & \mu_1^{(1)}, \mu_2^{(1)}, , \dots, \mu_1^{(\bar{k})}, \mu_2^{(\bar{k})}, \\ \text{right points}: \ & \lambda_1^{(1)}, \lambda_2^{(1)}, \dots, \lambda_1^{(\bar{k})}, \lambda_2^{(\bar{k})}. \end{aligned} \tag{4.32}$$

Next, the left and right interpolation points are grouped in multi-tuples for $j = 1, \dots, \tilde{k}$:

$$\boldsymbol{\mu}^{(j)} = \{(\mu_1^{(j)}),\ (\mu_1^{(j)},\ \mu_2^{(j)})\}, \quad \boldsymbol{\lambda}^{(j)} = \{(\lambda_1^{(j)}),\ (\lambda_2^{(j)},\ \lambda_1^{(j)})\}. \tag{4.33}$$

The generalized controllability matrix $\mathcal{R} \in \mathbb{C}^{n \times k}$ associated with the right multi-tuples $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \dots, \boldsymbol{\lambda}^{(\tilde{k})}$ is

$$\mathcal{R} = \left[\mathcal{R}^{(1)},\ \mathcal{R}^{(2)},\ \cdots,\ \mathcal{R}^{(\tilde{k})}\right], \tag{4.34}$$

where the matrices $\mathcal{R}^{(j)} \in \mathbb{C}^{n \times 2}$, $j = 1, \dots, \tilde{k}$, are associated with the j-th multi-tuple $\boldsymbol{\lambda}^{(j)}$ in (4.33) are given by

$$\mathcal{R}^{(j)} = \left[(\lambda_1^{(j)}\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{b}, \quad (\lambda_2^{(j)}\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{N}\,(\lambda_1^{(j)}\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{b},\right]. \tag{4.35}$$

Similarly, the generalized observability matrix $\mathcal{O} \in \mathbb{C}^{k \times n}$ associated with the left multi-tuples $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \dots, \boldsymbol{\mu}^{(\tilde{k})}$ is given by

$$\mathcal{O} = \left[\left(\mathcal{O}^{(1)}\right)^T,\ \left(\mathcal{O}^{(2)}\right)^T,\ \dots \left(\mathcal{O}^{(\bar{k})}\right)^T\right]^T \in \mathbb{C}^{k \times n}, \tag{4.36}$$

where $\mathcal{O}^{(i)} \in \mathbb{C}^{2 \times n}$, $i = 1, \dots, \bar{k}$, correspond to the i-th multi-tuple $\boldsymbol{\mu}^{(i)}$ in (4.33) and

$$\mathcal{O}^{(i)} = \begin{bmatrix} \mathbf{c}^T\,(\mu_1^{(i)}\mathbf{E} - \mathbf{A})^{-1} \\ \mathbf{c}^T\,(\mu_1^{(i)}\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{N}\,(\mu_2^{(i)}\mathbf{E} - \mathbf{A})^{-1} \end{bmatrix}. \tag{4.37}$$

Next, the Loewner matrix $\mathbb{L}$ and the shifted Loewner matrix $\mathbb{L}_s$ are defined using the generalized controllability (4.34) and observability (4.36) matrices as

$$\mathbb{L} = -\mathcal{O}\,\mathbf{E}\,\mathcal{R}, \quad \mathbb{L}_s = -\mathcal{O}\,\mathbf{A}\,\mathcal{R}. \tag{4.38}$$

The fact that the Loewner matrices are factorized in terms of the pairs of matrices $(\mathbf{E}, \mathbf{A})$ and $(\mathcal{O}, \mathcal{R})$ is an inherent property of the Loewner framework, which holds for both the bilinear [12] and quadratic-bilinear [68], and polynomial [5, 26] extensions of the method.

Analyzing the bilinear case and using the structure in (4.35) and in (4.37), it follows that:

$$\mathbb{L}^{(i,j)} = -\mathcal{O}^{(i)}\,\mathbf{E}\,\mathcal{R}^{(j)} = \begin{bmatrix} \frac{\mathbf{H}_1(\mu_1^{(i)}) - \mathbf{H}_1(\lambda_1^{(j)})}{\mu_1^{(i)} - \lambda_1^{(j)}} & \frac{\mathbf{H}_2(\mu_1^{(i)}, \lambda_1^{(j)}) - \mathbf{H}_2(\lambda_2^{(j)}, \lambda_1^{(j)})}{\mu_1^{(i)} - \lambda_2^{(j)}} \\ \frac{\mathbf{H}_2(\mu_1^{(i)}, \mu_2^{(i)}) - \mathbf{H}_2(\mu_1^{(i)}, \lambda_1^{(j)})}{\mu_2^{(i)} - \lambda_1^{(j)}} & \frac{\mathbf{H}_3(\mu_1^{(i)}, \mu_2^{(i)}, \lambda_1^{(j)}) - \mathbf{H}_3(\mu_1^{(i)}, \lambda_2^{(j)}, \lambda_1^{(j)})}{\mu_2^{(i)} - \lambda_2^{(j)}} \end{bmatrix} \tag{4.39}$$

and similarly for $\mathbb{L}_s{}^{(i,j)}$. Hence, matrices $\mathbb{L}$ and $\mathbb{L}_s$ are indeed data matrices since all their entries are samples of the system's regular transfer functions denoted with $\mathbf{H}_n^{\mathtt{reg}}(s_1,\ldots,s_n)$ (evaluated at the particular grid of points). Next, introduce matrices:

$$\mathbb{V} = \mathcal{O}\mathbf{b},\quad \mathbb{W} = \mathbf{c}\mathcal{R},\quad \mathbb{T} = \mathcal{O}\mathbf{N}\mathcal{R}, \tag{4.40}$$

Which can also be shown to be composed solely of data, i.e., evaluations of the regular transfer functions. For matrix $\mathbb{T}$, one can show that the (i,j) block is explicitly written as follows:

$$\mathbb{T}^{(i,j)} = \mathcal{O}^{(i)}\,\mathbf{N}\,\mathcal{R}^{(j)} = \begin{bmatrix} \mathbf{H}_2(\mu_1^{(i)},\lambda_1^{(j)}) & \mathbf{H}_3(\mu_1^{(i)},\lambda_2^{(j)},\lambda_1^{(j)}) \\ \mathbf{H}_3(\mu_1^{(i)},\mu_2^{(i)},\lambda_1^{(j)}) & \mathbf{H}_4(\mu_1^{(i)},\mu_2^{(i)},\lambda_2^{(j)},\lambda_1^{(j)}) \end{bmatrix}. \tag{4.41}$$

Remark 4.11 (What kind of data the bilinear Loewner framework needs?): The extension of the Loewner framework to the bilinear system class has been accomplished within the Volterra framework when the regular type of transfer functions Eq. (4.31) has been assumed. Therefore, the bilinear Loewner works using samples of Eq. (4.31) after direct numerical simulation (DNS). In that way, data are obtained that can be processed further within the bilinear Loewner framework, primarily for reduction or identification purposes. It would have been beneficial if there was a way to derive these measurements as samples of the regular type from physical experiments without assuming a known high-fidelity system that one wants to sample. ◊

Ideally, after simulating the system with particular inputs and collecting the corresponding outputs, we want to infer data directly from a physical measurement process. Under some mild assumptions, the nonlinear underline model can be modeled as a bilinear that Carleman's bilinearization scheme can approximate.

4.3.3. A challenging aspect in extending the Loewner framework

Consider the bilinear system in Eq. (4.28). As detailed in Section 4.3.1 and in [137], there are different ways of deriving generalized frequency response functions (GFRFs). The Loewner framework has been extended with the regulars for the bilinear case in [12]. Here, we would like to expose the fundamental issue for extending the Loewner framework with the symmetric type Volterra kernels. Next, once more, is the 2nd symmetric kernel for the bilinear system as in Eq. (4.28) that we can measure from the power spectrum after processing the time domain data as in [104]

$$H_2^{\mathtt{sym}}(s_1,s_2) = \frac{1}{2}\mathbf{C}\left((s_1+s_2)\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{N}\left((s_1\mathbf{E}-\mathbf{A})^{-1}\mathbf{B} + (s_2\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}\right). \tag{4.42}$$

To extend the Loewner framework for the bilinear case with the symmetric transfer functions, someone should start by appropriately defining the general observability $\mathcal{O}$ and controllability matrices $\mathcal{R}$. There are different ways of constructing these matrices, but here and only for illustration purposes, we introduce a particular choice with the following sets:

The general observability matrix

$$\mathcal{O}^{(i)} = \left[\begin{array}{c} \mathbf{c}^T(\mu_1^{(i)}\mathbf{E}-\mathbf{A})^{-1} \\ \mathbf{c}^T((\mu_1^{(i)}+\mu_2^{(i)})\mathbf{E}-\mathbf{A})^{-1}\mathbf{N}\left((\mu_2^{(i)}\mathbf{E}-\mathbf{A})^{-1}+(\mu_1^{(i)}\mathbf{E}-\mathbf{A})^{-1}\right) \end{array} \right], \tag{4.43}$$

and the general controllability matrix

$$\mathcal{R}^{(j)} = \left[\begin{array}{cc} (\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b} & ((\lambda_1^{(j)}+\lambda_2^{(j)})\mathbf{E}-\mathbf{A})^{-1}\mathbf{N}\left((\lambda_2^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}+(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}\right) \end{array} \right]. \tag{4.44}$$

The above choices seem reasonable, involving the bilinear matrix $\mathbf{N}$ into the controllability and observability counterparts. They will produce double-sided projection schemes as in the classical bilinear Loewner [12].

Similarly, as in Eq. (4.38), we want to derive the corresponding Loewner matrices (the Loewner pencil). We write the following 1st-1st blocked-wise multiplication:

$$\begin{aligned}
\mathbb{L}_{11} &= -\mathcal{O}_1^{(i)}\mathbf{E}\mathcal{R}_1^{(j)} = -\mathbf{c}^T(\mu_1^{(i)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{E}(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b} = \\
&= -\mathbf{c}^T(\mu_1^{(i)}\mathbf{E}-\mathbf{A})^{-1}\frac{\mu_1^{(i)}\mathbf{E}-\mathbf{A}-\lambda_1^{(j)}\mathbf{E}+\mathbf{A}}{\mu_1^{(i)}-\lambda_1^{(j)}}(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b} = \\
&= -\mathbf{c}^T(\mu_1^{(i)}\mathbf{E}-\mathbf{A})^{-1}\frac{(\mu_1^{(i)}\mathbf{E}-\mathbf{A})-(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})}{\mu_1^{(i)}-\lambda_1^{(j)}}(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b} = \\
&= \frac{-\mathbf{c}^T(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}+\mathbf{c}^T(\mu_1^{(i)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}}{\mu_1^{(i)}-\lambda_1^{(j)}} = \frac{H_1(\mu_1^{(i)})-H_1(\lambda_1^{(j)})}{\mu_1^{(i)}-\lambda_1^{(j)}}.
\end{aligned} \tag{4.45}$$

The importance of the above result is that the Loewner matrix can be directly constructed from measurements of H_1 that can be measured from the power spectrum as the 1st kernel that drives the 1st harmonic. We continue with the 1st-2nd blocked-wise multiplication:

$$\begin{aligned}
&\mathbb{L}_{12} = -\mathcal{O}_1^{(i)}\mathbf{E}\mathcal{R}_2^{(j)} = \\
&= -\mathbf{c}^T(\mu_1^{(i)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{E}\left((\lambda_1^{(j)}+\lambda_2^{(j)})\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{N}\left((\lambda_2^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}+(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}\right) = \\
&= -\mathbf{c}^T(\mu_1^{(i)}\mathbf{E}-\mathbf{A})^{-1}\frac{\mu_1^{(i)}\mathbf{E}-\mathbf{A}-(\lambda_1^{(j)}+\lambda_2^{(j)})\mathbf{E}-\mathbf{A}}{\mu_1^{(i)}-(\lambda_1^{(j)}+\lambda_2^{(j)})}\left((\lambda_1^{(j)}+\lambda_2^{(j)})\mathbf{E}-\mathbf{A}\right)^{-1}\times\cdots \\
&\cdots\times\mathbf{N}\left((\lambda_2^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}+(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}\right) = \\
&= \frac{\mathbf{c}^T\left(\mu_1^{(i)}\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{N}\left((\lambda_2^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}+(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}\right)-H_2^{\mathtt{sym}}(\lambda_1^{(i)},\lambda_2^{(j)})}{\mu_1^{(i)}-(\lambda_1^{(j)}+\lambda_2^{(j)})}.
\end{aligned} \tag{4.46}$$

As it is evident, the Loewner block $\mathbb{L}_{12}$ contains the 2nd symmetric transfer function $H_2^{\mathtt{sym}}$ that can be measured from the power spectrum as the 2nd kernel that drives the 2nd harmonic, but the extra entry

$$H_2^{(?)}(\mu_1^{(i)},\lambda_1^{(j)},\lambda_2^{(j)}) = \mathbf{c}^T\left(\mu_1^{(i)}\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{N}\left((\lambda_2^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}+(\lambda_1^{(j)}\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}\right), \tag{4.47}$$

is a generalized kernel that cannot be directly associated with measurements from the power spectrum when considering time domain data. Thus, with the above construction of the generalized observability/controllability matrices, we obtain Loewner matrices. However, filling these Loewner matrices with actual data (processing the time domain) is impossible as long as the entries of the Loewner matrix are inconsistent with the symmetric generalized frequency response structure. Another approach is constraining the bilinear operator to be symmetric and appropriately constructing the controllability and observability matrices. Thus, measuring the Loewner pencil from time domain data after processing them to the frequency domain is possible. However, the assumption that $\mathbf{N}$ is a symmetric matrix may be restrictive. Therefore, we proceed in this thesis keeping the $\mathbf{N}$ unconstrained.

4.3.4. Pole-residue form of the bilinear regular Volterra kernels

Starting with the bilinear system in Eq. (4.28), we can enforce the canonical form of the linear subsystem. That is

$$\breve{\mathbf{A}} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}, \; \breve{\mathbf{c}}^T = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \; \breve{\mathbf{b}} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, \; \breve{\mathbf{N}} = \begin{bmatrix} \eta_{11} & \cdots & \eta_{1n} \\ \vdots & \ddots & \vdots \\ \eta_{n1} & \cdots & \eta_{nn} \end{bmatrix}. \tag{4.48}$$

For simplicity, we assume that the linear subsystem of the bilinear model in (4.48) is minimal, i.e., $c_i, b_j \neq 0$ for all $1 \leq i, j \leq n$. In this case, we can say that the poles of the linear model $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ perfectly coincide with the eigenvalues of matrix $\mathbf{A}$.

The first transfer function in the pole-residue form is $H_1(s_1) = \sum_{i=1}^{n} \frac{\gamma_i}{s_1 - \lambda_i}$. Following [55], the pole residue forms of the higher regular kernels are

$$\begin{aligned} H_2^{\mathtt{reg}}(s_1, s_2) &= \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{c_i \eta_{i,j} b_j}{(s_2 - \lambda_i)(s_1 - \lambda_j)} \\ &\vdots \\ H_m^{\mathtt{reg}}(s_1, \ldots, s_m) &= \sum_{i_1=1}^{n} \cdots \sum_{i_m=1}^{n} \frac{c_{i_1} \eta_{i_1,i_2} \cdots \eta_{i_{m-1},i_m} b_{i_m}}{(s_m - \lambda_{i_m}) \cdots (s_2 - \lambda_{i_2})(s_1 - \lambda_{i_1})}. \end{aligned} \tag{4.49}$$

4.3.5. Pole-residue form of the bilinear symmetric Volterra kernels

Similarly to the variational approach described in [12] that helps the explicit derivation of regulars/triangular kernels, the growing exponential approach [137] constructs symmetric type Volterra kernels that yield a Volterra series. Moreover, those symmetric kernels comprise the appropriate set that connects with the physical measurement setup. We assume the following multi-harmonic complex signal with normalized amplitudes equal with the unit and frequencies $s_i = \mathrm{j}\omega_i,\ i = 1, \ldots, m$. The following system excitation is also known as the probing method.

$$u(t) = \sum_{i=1}^{k} e^{s_i t} \tag{4.50}$$

The growing exponential approach: Any nonlinear system that yields a convergent Volterra series, after excitation with input as in Eq. (4.50) has the following steady-state response for $M \to \infty$

$$\begin{aligned} \mathbf{x_{ss}}(t) &= \sum_{m=1}^{M} \sum_{i_1=1}^{k} \cdots \sum_{i_k=1}^{k} \tilde{\mathbf{G}}_m(s_{i_1}, \ldots, s_{i_k}) e^{(s_{i_1}+\cdots+s_{i_k})t} = \\ &= \sum_{m=1}^{M} \sum_{p(m)=1} \mathbf{G}_{p_1(m)\cdots p_k(m)}(s_1, \ldots, s_k) e^{(p_1(m)s_1+\cdots+p_k(m)s_k)t} \end{aligned} \tag{4.51}$$

where $\sum_{p(m)}$ indicates a k-fold sum over all integer indices $p_1(m), \ldots, p_k(m)$ such that $0 \leq p_i(m) \leq m,\ p_1(m) + \cdots + p_k(m) = m$ and

$$\mathbf{G}_{p_1(m)\cdots p_k(m)}(s_1, \ldots, s_k) = \frac{m!}{p_1(m)! \cdots p_k(m)!} \tilde{\mathbf{G}}_m \left(\underbrace{s_1, \ldots, s_1}_{p_1(m)}, \ldots, \underbrace{s_k, \ldots, s_k}_{p_k(m)} \right). \tag{4.52}$$

Remark 4.12 (Input-output mapping with the symmetric Volterra kernels):
After multiplication with the observability vector $\mathbf{C}$ from the left on the above relations, the steady-state input-output response is

$$\begin{aligned} y_{ss}(t) = \mathbf{C}\mathbf{x}_{ss}(t) &= \sum_{m=1}^{M} \sum_{p(m)=1} \mathbf{c}\mathbf{G}_{p_1(m)\cdots p_k(m)}(s_1, \ldots, s_k) e^{(p_1(m)s_1+\cdots+p_k(m)s_k)t} \\ &= \sum_{m=1}^{M} \sum_{p(m)=1} H_{p_1(m)\cdots p_k(m)}(s_1, \ldots, s_k) e^{(p_1(m)s_1+\cdots+p_k(m)s_k)t} \end{aligned} \tag{4.53}$$

where H_m is the input-output symmetric generalized frequency response function. ♢

Remark 4.13 (Symmetric kernel derivation):
As evident from the relations in Eqs. (4.51) and (4.53), identifying the mth symmetric kernel needs excitation with at least an mth harmonic input. ♢

The derivation of the symmetric transfer function becomes tedious with the above analysis and does not yield a concise way of representing these extended expressions. Therefore, as it is at the heart of our problem, we derive formulas that concisely connect the symmetric transfer functions with the other types. In [137], the relations between the regular, triangular, and symmetric are:

$$\begin{aligned} H_m^{\texttt{sym}}(s_1,s_2,\ldots,s_m) &= \frac{1}{m!}\sum_{p(\cdot)} H_m^{\texttt{tri}}(s_{p(1)},s_{p(2)},\ldots,s_{p(m)}) \\ H_m^{\texttt{tri}}(s_1,s_2,\ldots,s_m) &= H_m^{\texttt{reg}}(s_1,s_1+s_2,\ldots,s_1+\cdots+s_m) \end{aligned} \tag{4.54}$$

With a bit of inspection, the connection between the symmetric and the regular type is

$$H_m^{\texttt{sym}}(s_1,s_2,\ldots,s_m) = \frac{1}{m!}\sum_{p(\cdot)} H_m^{\texttt{reg}}(s_{p(1)},s_{p(1)}+s_{p(2)},\ldots,s_{p(1)}+s_{p(2)}+\cdots+s_{p(m)}) \tag{4.55}$$

where the $p(\cdot)$ denotes all the $m!$ permutations of the frequency arguments $s_1,\ldots,s_m$.

4.3.6. Connecting the symmetric with regular Volterra kernels

The pole-residue form for the symmetric generalized frequency response functions is

$$H_m^{\texttt{sym}}(s_1,s_2,\ldots,s_m) = \frac{1}{m!}\sum_{p(\cdot)}\sum_{i_1=1}^{n}\cdots\sum_{i_m=1}^{n}\frac{c_{i_1}\eta_{i_1,i_2}\cdots\eta_{i_{m-1},i_m}b_{i_m}}{(s_{p(m)}+\cdots+s_{p(1)}-\lambda_{i_m})\cdots(s_{p(1)}-\lambda_{i_1})} \tag{4.56}$$

where $p(m)$ denotes all the $m!$ permutations of the index set $1,\ldots,m$.

Identifying the symmetric pole-residue from measurements From a data-driven perspective, the pole residue form of the symmetric generalized frequency response function in Eq. (4.56) has the following properties. The left-hand-side consist of a measurement at $(s_1,s_2,\ldots,s_m)$ and the right-hand-side contains the following unknowns; 1) The order n 2) The residues $\rho_{i_1,i_2,\ldots,i_m}$ and 3) the singularities $\lambda_i,\ i=1,\ldots,n$. Estimating the previous unknowns is a nonlinear optimization problem. We solve this problem in two steps. The first step is to estimate the λ_i, and the second is to compute the residues.

Computing the singular points λ_i. By equating $s_1=s_2=\cdots=s_m=s$, the pole-residue form remains as:

$$\begin{aligned} f_m(s) = H_m^{\texttt{sym}}(s,s,\ldots,s) &= \sum_{i_1=1}^{n}\cdots\sum_{i_m=1}^{n}\frac{c_{i_1}\eta_{i_1,i_2}\cdots\eta_{i_{m-1},i_m}b_{i_m}}{(ms-\lambda_{i_m})\cdots(2s-\lambda_{i_2})(s-\lambda_{i_1})} \Leftrightarrow \\ f_m(s) = H_m^{\texttt{sym}}(s,s,\ldots,s) &= \frac{1}{m!}\sum_{i_1=1}^{n}\cdots\sum_{i_m=1}^{n}\frac{c_{i_1}\eta_{i_1,i_2}\cdots\eta_{i_{m-1},i_m}b_{i_m}}{(s-\lambda_{i_m}/m)\cdots(s-\lambda_{i_2}/2)(s-\lambda_{i_1})} \end{aligned} \tag{4.57}$$

Lemma 4.14 (Intrusive analysis):
Consider $(\mathbf{A},\mathbf{N},\mathbf{b},\mathbf{c})$ a bilinear system, and λ_i's be the singular points (e.g., poles in the linear case) of the linear sub-system $(\mathbf{A},\mathbf{b},\mathbf{c})$, for $1\leq i\leq n$ (i.e., the eigenvalues of the matrix $\mathbf{A}$). Then, the poles of the mth symmetric generalized frequency response function $f_m(s)=H_m(s,\ldots,s)$ are $m\times n$ in total and can be written explicitly as all the permutations of the following vector with length m as $\begin{bmatrix}\lambda_{i_m}/m & \cdots & \lambda_{i_2}/2 & \lambda_{i_1}\end{bmatrix}$ with indexes as $i_{(\cdot)}=1,\ldots,n$. ◊

Considering the above result non-intrusively, we can collect measurements of the symmetric generalized transfer functions $H_m^{\mathtt{sym}}(s,\ldots,s)$ and interpret them as bilinear symmetric GFRFs. Then, all the recovered r poles ($r \leq n$) of the mth symmetric generalized frequency response function $f_m(s) = H_m^{\mathtt{sym}}(s,\ldots,s)$ are $m \times r$ in total and can be written explicitly as all the permutations of the following vector with length m as $\left[\begin{array}{cccc} \lambda_{i_m}/m & \cdots & \lambda_{i_2}/2 & \lambda_{i_1}\end{array}\right]$ with indices as $i_{(\cdot)} = 1,\ldots,r$.

The difference between the original order n and the recovered order $r \leq n$ may be explained since the controllability/observability matrices are not of full rank [12]. As a result, some poles cannot be detectable, and the corresponding residue remains zero. In general, when the minimality of the bilinear system is different from the linear subsystem, higher generalized transfer functions should be involved for identification.

By having measurements on the diagonal of the symmetric generalized frequency response functions $f_m(s) = H_m(s,\ldots,s)$, we can apply the linear Loewner framework Chapter 3 to recover the univariate rational function $f_m(s)$ and interpret it as the transfer function of a linear system with the realization $(\hat{\mathbf{A}}_m, \hat{\mathbf{b}}_m, \hat{\mathbf{c}}_m)$. Thus, the poles at level m can be computed as the eigenvalues of the linear operator $\mu_i^{(m)} = eig(\hat{A}_m)$, $i = 1,\ldots,\ell_m \leq mn$. As we have the ℓ recovered poles of the mth level transfer function, we must assign them in $m!\ell_m^m$ positions. It is to be noted that this number could be reduced as we can detect splittings of the original poles and assign them to the appropriate positions.

By having the poles $\mu_i^{(m)}$ with cardinality ℓ_m, the pole-residue form of the symmetric generalized frequency response function takes the form

$$\begin{aligned}&\hat{H}_m^{\mathtt{sym}}(s_1,s_2,\ldots,s_m) = \\ &= \frac{1}{m!}\sum_{p(\cdot)}\sum_{i_1=1}^{\ell_m}\cdots\sum_{i_m=1}^{\ell_m}\frac{\rho_{i_1,i_2,\ldots,i_m}}{(s_{p(m)}+\cdots+s_{p(1)}-\mu_{i_m})\cdots(s_{p(2)}+s_{p(1)}-\mu_{i_2})(s_{p(1)}-\mu_{i_1})}\end{aligned} \tag{4.58}$$

where the remaining unknowns are the residues only. Therefore, by having measurements of $H_m^{\mathtt{sym}}$, we can solve linearly for the residues. To simplify the formula above, we introduce the following notation of the symmetric base

$$\begin{aligned}&\mathcal{B}^{\mathtt{sym}}(s_1,\ldots,s_m;\mu_{i_1,\ldots,i_m}) = \\ &= \frac{1}{m!}\sum_{p(\cdot)}\frac{1}{(s_{p(m)}+\cdots+s_{p(1)}-\mu_{i_m})\cdots(s_{p(2)}+s_{p(1)}-\mu_{i_2})(s_{p(1)}-\mu_{i_1})}\end{aligned} \tag{4.59}$$

where the symmetric mth generalized frequency response function is

$$\hat{H}_m^{\mathtt{sym}}(s_1,s_2,\ldots,s_m) = \sum_{i_1=1}^{\ell_m}\cdots\sum_{i_m=1}^{\ell_m}\mathcal{B}_m^{\mathtt{sym}}(s_1,\ldots,s_m;\mu_{i_1},\ldots,\mu_{i_m})\rho_{i_1,i_2,\ldots,i_m}, \tag{4.60}$$

where the residues $\rho_{i_1,\ldots,i_m}$ are to be determined. By setting $\mathbf{s}_m^{(k)} = (s_1^{(k)}, s_2^{(k)},\ldots,s_m^{(k)})$ the mth-dimensional argument at the kth measurement, we can write

$$\hat{H}_m^{\mathtt{sym}}(\mathbf{s}_m^{(k)}) = \sum_{i_1=1}^{\ell_m}\cdots\sum_{i_m=1}^{\ell_m}\mathcal{B}_m^{\mathtt{sym}}(\mathbf{s}_m^{(k)};\mu_{i_1},\ldots,\mu_{i_m})\rho_{i_1,i_2,\ldots,i_m}, \tag{4.61}$$

where, the scalar complex number at the kth measurement $\hat{H}_m^{\mathtt{sym}}(\mathbf{s}_m^{(k)})$ can be decomposed in a matrix product as

$$\underbrace{\begin{bmatrix} \mathcal{B}_m^{\mathtt{sys}}(\mathbf{s}_m^{(k)};\mu_{1,1,\ldots,1}) & \mathcal{B}_m^{\mathtt{sys}}(\mathbf{s}_m^{(k)};\mu_{1,1,\ldots,2}) & \cdots & \mathcal{B}_m^{\mathtt{sys}}(\mathbf{s}_m^{(k)};\mu_{\ell_m,\ell_m,\ldots,\ell_m}) \end{bmatrix}}_{\mathcal{M}^{(k)}\in\mathbb{C}^{1\times\ell_m^m}} \underbrace{\begin{bmatrix} \rho_{1,1,\ldots,1} \\ \rho_{1,1,\ldots,2} \\ \vdots \\ \rho_{\ell_m,\ell_m,\ldots,\ell_m} \end{bmatrix}}_{\boldsymbol{\rho}}, \tag{4.62}$$

and after appending k measurements, the least squares linear problem is defined with the following matrices.

$$\begin{aligned} \mathcal{M} &= \begin{bmatrix} \mathcal{B}_m^{\mathtt{sym}}(\mathbf{s}_m^{(1)};\mu_{1,1,\ldots,1}) & \mathcal{B}_m^{\mathtt{sym}}(\mathbf{s}_m^{(1)};\mu_{1,1,\ldots,2}) & \cdots & \mathcal{B}_m^{\mathtt{sym}}(\mathbf{s}_m^{(1)};\mu_{\ell_m,\ell_m,\ldots,\ell_m}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{B}_m^{\mathtt{sym}}(\mathbf{s}_m^{(k)};\mu_{1,1,\ldots,1}) & \mathcal{B}_m^{\mathtt{sys}}(\mathbf{s}_m^{(k)};\mu_{1,1,\ldots,2}) & \cdots & \mathcal{B}_m^{\mathtt{sym}}(\mathbf{s}_m^{(k)};\mu_{\ell_m,\ell_m,\ldots,\ell_m}) \end{bmatrix} \in \mathbb{C}^{k\times\ell_m^m}, \\ \mathcal{L} &= \begin{bmatrix} H_m^{\mathtt{sym}}(\mathbf{s}_m^{(1)}) \\ \vdots \\ H_m^{\mathtt{sym}}(\mathbf{s}_m^{(k)}) \end{bmatrix} \in \mathbb{C}^{k\times 1},\ \boldsymbol{\rho} = \begin{bmatrix} \rho_{1,1,\ldots,1} \\ \rho_{1,1,\ldots,2} \\ \vdots \\ \rho_{\ell_m,\ell_m,\ldots,\ell_m} \end{bmatrix} \in \mathbb{C}^{\ell_m^m\times 1},\ \text{where } \mathcal{M}\boldsymbol{\rho} = \mathcal{L}. \end{aligned} \tag{4.63}$$

Lemma 4.15 (Identification of the pole residue form):
The base that is spanned in Eq. (4.59) with the measurements $\mathbf{s}_m^{(k)} = \left(s_1^{(k)}, s_2^{(k)}, \ldots, s_m^{(k)}\right)$, it constructs a full rank column space for the least squares matrix $\mathcal{M} \in \mathbb{C}^{k\times\ell_m^m}$. When $k \geq \ell_m^m$, it holds that $\mathtt{rank}(\mathcal{M}) = \ell_m^m$, and the vector solution with the ℓ_m^m unknown residues from $\boldsymbol{\rho}= \mathcal{M}^{-1}\mathcal{L}$ is unique and well defined. $\diamondsuit$

Remark 4.16 (Data reduction):
For solving the system above, due to the symmetric nature of the symmetric generalized transfer function $H_m(s_1, s_2, \ldots, s_m)$, for a single measurement $\mathbf{s}_m^{(k)} = \left(s_1^{(k)}, s_2^{(k)}, \ldots, s_m^{(k)}\right)$, we have in total $m!$ measurements. That happens as all the possible permutations in the argument provide the same complex value. $\diamondsuit$

Remark 4.17 (Computational effort):
To infer the *mth* regular transfer function from the *mth* symmetric transfer function with ℓ_m the number of singular points at the *mth* level transfer function, a linear system of dimension ℓ_m^m should be solved. For the Burgers' example of dimension $n = 6$, the 4th kernel has $\ell_4 = 10$ singular points. Therefore, one would need $\ell_4^4 = 10^4 = 10,000$ measurements of the symmetric transfer function $H_4^{\mathtt{sym}}(s_1, s_2, s_3, s_4)$ so to compute the residues and the pole reside form of the transfer function. $\diamondsuit$

By having the symmetric *mth* kernel in the pole residue form, we know the singularities λ_i and the residues ρ_i; we can infer the regular pole residue kernel by assigning the frequency variable $s_1, s_2, \ldots, s_m$ as in the following way:

$$
\begin{aligned}
H_2^{\text{sym}}(s_1, s_2) &= \frac{1}{2}\sum_{i=1}^{r}\sum_{i=1}^{r} \frac{\rho_{i,j}}{(s_1+s_2-\lambda_i)(s_1-\lambda_j)} + \frac{\rho_{i,j}}{(s_1+s_2-\lambda_i)(s_2-\lambda_j)} \\
&= \underbrace{\frac{1}{2}\sum_{i=1}^{r}\sum_{i=1}^{r} \frac{\rho_{i,j}}{(s_1+s_2-\lambda_i)(s_1-\lambda_j)}}_{H_2^{\texttt{reg}}(s_1, s_1+s_2)} + \underbrace{\frac{1}{2}\sum_{i=1}^{r}\sum_{i=1}^{r} \frac{\rho_{i,j}}{(s_1+s_2-\lambda_i)(s_2-\lambda_j)}}_{H_2^{\texttt{reg}}(s_2, s_1+s_2)}.
\end{aligned}
$$

Finally, by setting $(s_1 + s_2) \leftarrow s_2$

$$
H_2^{\texttt{reg}}(s_1, s_2) = \sum_{i=1}^{r}\sum_{j=1}^{r} \frac{\rho_{i,j}}{(s_2-\lambda_i)(s_1-\lambda_j)}.
$$

Remark 4.18 (Connecting the bilinear Loewner approach [12] with the time-domain): The procedure is generalizable for identifying higher kernels H_n in the regular type. Therefore, a connection with the bilinear Loewner framework has been accomplished. ◊

4.3.7. Numerical examples

4.3.7.1. A toy example

Example 4.3 (A toy example of order 2):
Consider the following bilinear system: bilinear minimal order 2 and linear minimal order 1.

$$
\mathbf{A} = \begin{pmatrix} -1.0 & 0 \\ 0 & -0.1 \end{pmatrix},\ \mathbf{N} = \begin{pmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1.0 \\ 0 \end{pmatrix},\ \mathbf{c}^T = \begin{pmatrix} 1.0 \\ 0 \end{pmatrix}. \tag{4.64}
$$

◊

The characteristic of the above bilinear system is that it has a bilinear minimal order of 2, but the linear subsystem has a minimal order of 1. The Loewner decays, along with the recovered poles, are presented in Fig. 4.8: Assigning the poles with the previous

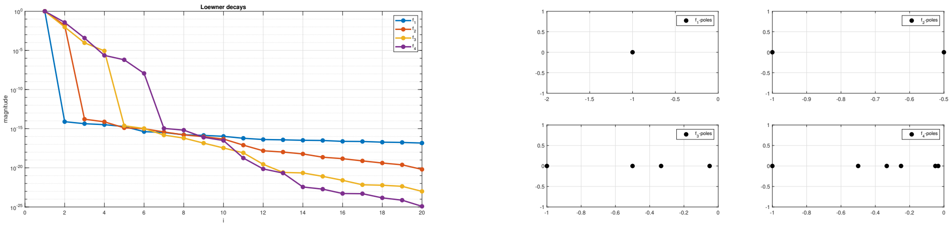


Figure 4.8.: The decay of the Loewner matrices (left). Recovered singular points (e.g., poles).

methodology, we can infer the regular kernels from the symmetric measurements and apply a bilinear Loewner. Next are the estimations over the diagonals of the higher kernels that recovered with the Loewner framework. Constructing the bilinear system of order 2 in the time domain, we obtain the following results in Fig. 4.9. Finally,

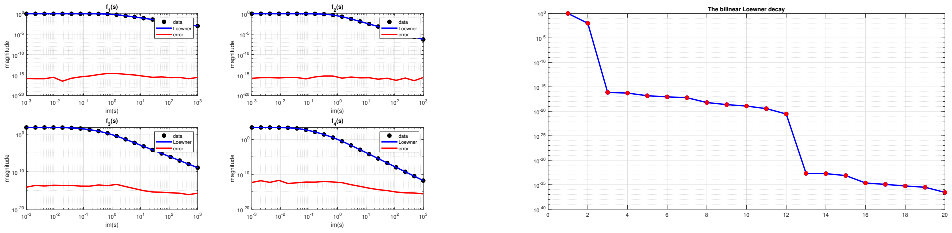


Figure 4.9.: Generalized transfer functions recovery and singular value decay of the bilinear Loewner matrix.

after inferring the regulars and applying the bilinear Loewner framework, the bilinear Loewner decay can correctly detect the minimal bilinear order and results in the bilinear identification of the minimal bilinear model as the normalized error is close to machine precision in Fig. 4.10.

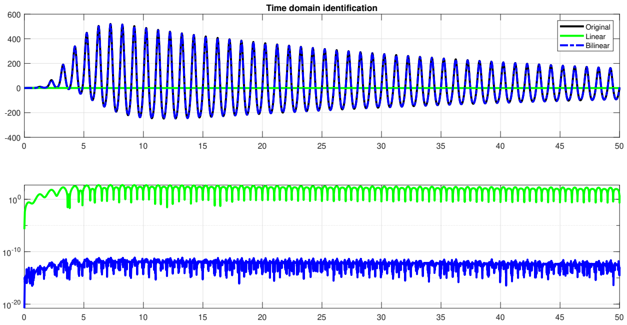


Figure 4.10.: Bilinear identification from i/o harmonic time domain data with bilinear Loewner.

4.3.7.2. Burgers' example with low dimension

Example 4.4 (The viscous Burgers' equation):
This example illustrates the proposed time domain bilinear realization method using the developed bilinear Loewner from [12] after inferring the regular kernels from the measured symmetric kernels. We consider the viscus Burgers' equation with dimension $n = 6$ and viscosity parameter $\nu = 0.001$. Another characteristic of this example is that it is not a minimal bilinear system of order $n = 6$. The minimality can be checked from Isidori's

realization algorithm in [95], and the minimal order of the bilinear system is equal to 4.

$$\mathbf{A} = \begin{pmatrix} -0.018 & 0.009 & 0 & -0.75 & -0.75 & 0 \\ 0.009 & -0.018 & 0 & 0.75 & 0.75 & 0 \\ 0 & 0 & -0.036 & 0.009 & 0.009 & 0 \\ 0 & 0 & 0.009 & -0.036 & 0 & 0.009 \\ 0 & 0 & 0.009 & 0 & -0.036 & 0.009 \\ 0 & 0 & 0 & 0.009 & 0.009 & -0.036 \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} 0.009 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \ \mathbf{C}^T = \begin{pmatrix} 0.5 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and the bilinear operator

$$\mathbf{N} = \begin{pmatrix} 1.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.018 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.009 & 0 & 0 & 0 & 0 \\ 0 & 0.009 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We start our analysis by getting the poles from the diagonal of the generalized frequency response functions $f_m(s) = H_m(\underbrace{s, \ldots, s}_{m \text{ times}})$. The equivalent of this over the actual measurement setup is to use a single harmonic input and, after kernel (harmonic) separation Section 2.3.2 to estimate the higher harmonics.

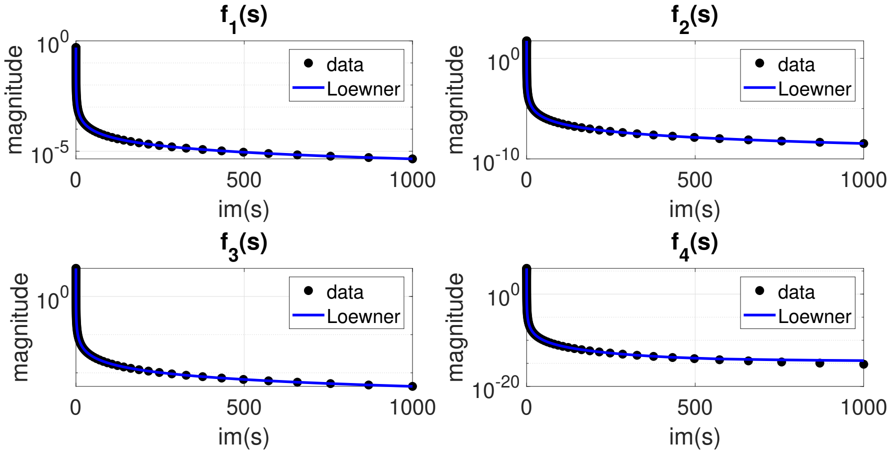


Figure 4.11.: Identify the first four higher generalized transfer functions over the diagonal $H_n(s, \ldots, s)$ with the Loewner framework.

By estimating (small perturbations are neglected) the poles as $\mu_{i_m}^{(m)} = eig(\hat{\mathbf{A}}_m)$, we get

- For $m = 1$, $\mu_{i_1}^{(1)} = \texttt{eig}(\hat{\mathbf{A}}_1) = -0.009$
- For $m = 2$, $\mu_{i_2}^{(2)} = \texttt{eig}(\hat{\mathbf{A}}_2) = \begin{pmatrix} -0.027 & -0.0045 & -0.009 \end{pmatrix}$

- For $m = 3$, $\mu_{i_3}^{(3)} = \texttt{eig}(\hat{\mathbf{A}}_3) = \begin{pmatrix} -0.027 & -0.027 & -0.0135 & -0.009 & -0.009 & -0.0045 & -0.003 \end{pmatrix}$
- For $m = 4$, $\mu_{i_4}^{(4)} = \texttt{eig}(\hat{\mathbf{A}}_4) =$ $\begin{pmatrix} -0.027 & -0.027 & -0.0135 & -0.009 & -0.009 & -0.009 & -0.006 & -0.0045 & -0.003 & -0.00225 \end{pmatrix}$ $\Diamond$

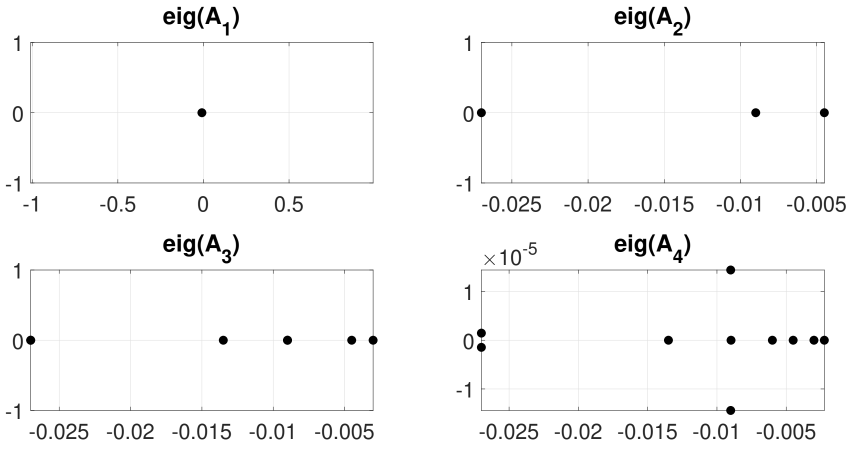


Figure 4.12.: Singular points recovery and small perturbations.

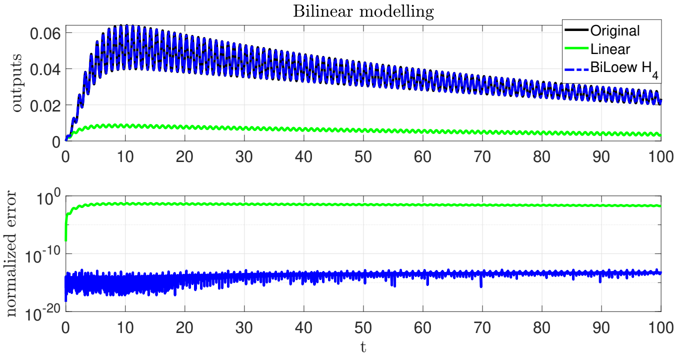


Figure 4.13.: Identification of the Burgers' bilinear model from time domain data with four kernels H_1, H_2, H_3, H_4.

4.3.7.3. Burgers' equation of large-dimension

Example 4.5 (A weakly nonlinear example of large-dimension):
Consider the Burgers' equation model of order 420 with a viscosity parameter of 0.001. We take measurements up to the 2nd harmonic by assuming that higher kernels (i.e., higher harmonics) are negligible, and the assumption of setting those equal to zero is realistic.

Realization algorithms exist based on the Loewner framework for the first two symmetric kernels that can be measured. Therefore, we can derive explicit rational function over the 2D domain of definition $(s_1, s_2) \in \mathbb{C}^2$. These are $H_1(s_1)$ and $H_2^{\mathtt{sym}}(s_1, s_2)$.

To infer the poles (singular points - roots of the denominator) of these two symmetric functions, we set $s_1 = s_2 = s$ and apply 1D Loewner. The decays of the singular values of the univariate functions $f_1(s) = H_1(s)$ and $f_2(s) = H_2^{\mathtt{sym}}(s, s)$ are illustrated in the following figure Fig. 4.14.

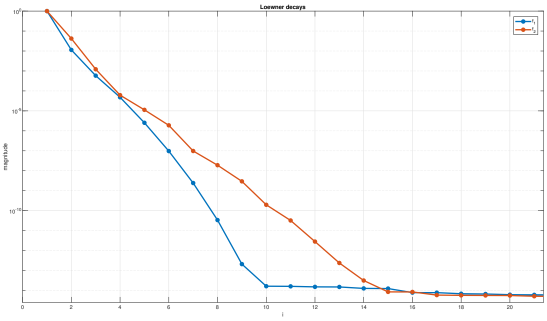


Figure 4.14.: Decay of the singular values for the first two diagonals of the kernels H_1, $H_2^{\mathtt{sym}}$.

From the above decays, we can compute the poles of $f_1(s)$ with order $r = 9$ and of $f_2(s)$ with order $r = 14$, along with the approximation results over the diagonals. These are displayed in the following figure Fig. 4.15.

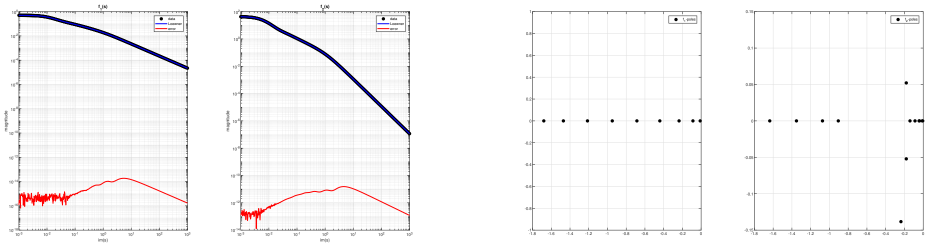


Figure 4.15.: Left: Kernel estimation on the diagonals. Right: Identified poles.

Now, with the discovered set of poles, we can construct the symmetric bases for H_1, $H_2^{\mathtt{sym}}$, and by solving linear systems, we can get the residues. After inferring the identical residues over the regular bases, we obtain the regular transfer functions $H_1(s)$, $H_2^{\mathtt{reg}}(s_1, s_2)$.

Using these two discovered regular functions and assuming all the higher kernels are negligible, we have the following bilinear Loewner decay Fig. 4.16(left). Finally, in Fig. 4.16(right), comparing the original large-scale model with the reduced bilinear indicates a significant improvement when a bilinear model is constructed. ◊

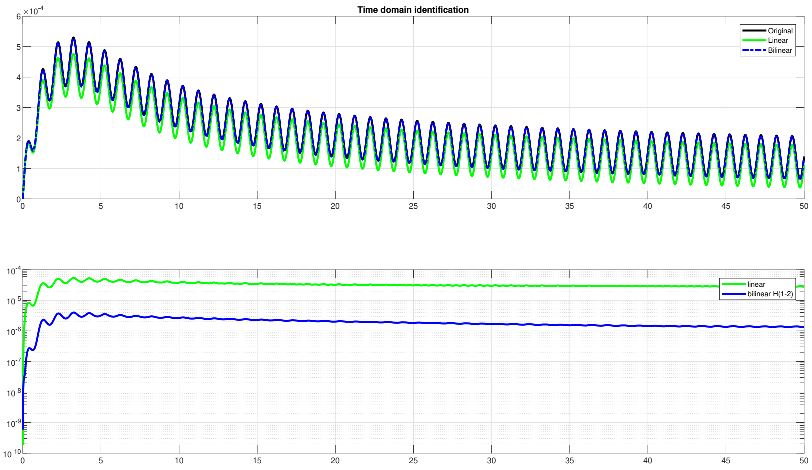


Figure 4.16.: The bilinear Loewner with zero padding ($H_3 = H_4 = 0$) allows a truncation with order 16. The reduced bilinear model is stable and improves the performance fit by two magnitudes compared to the linear approximant.

Remark 4.19 (Accuracy of the recovered poles and Jordan blocks):
The recovered poles, especially for the higher kernels $H_n(s,\ldots,s)$, are not accurately captured from the 1D Loewner. The reason is that the lifted bilinear symmetric kernels contain Jordan blocks after setting $s_1 = s_2 = \ldots = s_n = s$. Therefore, the poles are not "simple" with algebraic multiplicity equal to one. Consequently, the numerical precision of detecting such poles drops to the square root of the machine's numerical precision. The current machine precision with respect to IEEE is $\epsilon \approx 1e-16$ and drops to $\sqrt{\epsilon} \approx 1e-8$.$\diamondsuit$

The proposed time bilinear Loewner is a straightforward theoretical extension from the frequency bilinear Loewner in [12]. This was possible after theoretically connecting the different types of kernels within the Volterra framework. In particular, the bilinear Loewner framework needs regular kernel measurements, where time-domain measurements can only be processed from the symmetric type.

At the heart of the proposed method is the process of identifying the symmetric kernels after solving a nonlinear optimization problem in two steps;

1. detect the singularities (poles) of the higher kernels with linear Loewner (1D Loewner) and

2. Solve a linear system to compute the residues. By having the poles and residues of the symmetric kernels, i.e., the pole residue form of the symmetric kernels, with only a simple variable change assigning, we can identify the regular kernels. Finally, by having the regulars, the developed bilinear framework [12] can identify minimal bilinear systems (without coinciding with the linear minimality) or construct good bilinear models.

Due to the complexity of the number of poles for the higher kernels, we illustrated that the method could perform well when higher harmonics (kernels) can be assumed zero (zero padding). This stands in accordance with the Volterra series approximation, where due to the polynomial structure, the convergence of the Volterra series relies on the concept of weakly nonlinear dynamics. Another way to avoid the numerical issues that arise by applying the Linear framework on the diagonals to infer the poles is using the parametric Loewner framework [116, 12]. This can be done by asserting that $H_2^{\mathtt{sym}}(s_1, s_2)$ can be realized as $H(s, p)$. After realizing the $H_2^{\mathtt{sym}}(s_1, s_2)$, still there is no obvious way of getting the $H_2^{\mathtt{reg}}(s_1, s_2)$. We must compute the poles or the 2D singularities to regain this connection. We noticed the previous issue with the Jordan blocks after setting $s = s_1 = s_2$. Thus, an alternative is to employ multivariate polynomial factorization with partial differential equations [60]. Theoretically, this can avoid the numerical issues of detecting the 2D singularities, e.g., $(s_1+s_2+1)(s_1+1)(s_2+1)$ after setting $s = s_1 = s_2$ where results to a denominator with 1 not to be a simple pole, i.e., $(2s + 1)(s + 1)^2$. Combining multi-parametric Loewner with multi-variable polynomial factorization could be beneficial in tackling this problem. Still, we leave this research direction open for future research endeavors.

4.4. Spectral bilinear identification from state access

So far, the methods presented in the chapter work toward reducing or identifying bilinear systems from input-output time-domain data. They construct input invariant models. On the other hand, snapshot-based methods such as POD, DMD, and OpInf construct models with input bias, where these models can perform well for inputs close to the training one.

We present a snapshot-based method inspired by the classical concept of sampling state snapshots, but in addition, we achieve identification. As mentioned earlier, the step forward from the well-known snapshot-based methods is that we successfully construct models that stay input invariant.

Consider the continuous bilinear system of order n that is described in the following ODE

$$\mathbf{\Sigma}_b : \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{N}\mathbf{x}(t)u(t) + \mathbf{B}u(t),\ \mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0}. \tag{4.65}$$

With $s_i \in \mathbb{C},\ i = 1, \ldots, m$, the general mth-tone complex input is defined as

$$u(t) = \sum_{i=1}^{m} e^{s_i t} = e^{s_1 t} + e^{s_2 t} + \cdots + e^{s_m t}, \tag{4.66}$$

and with the growing exponential approach, the steady-state solution can be assumed as

$$\mathbf{x}_{ss}(t) = \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \mathbf{G}_{(k_1+k_2+\cdots+k_m)}(k_1 s_1, k_2 s_2, \ldots, k_m s_m) e^{(k_1 s_1 + k_2 s_2 + \cdots + k_m s_m)t}. \tag{4.67}$$

Remark 4.20 (The mth symmetric Volterra kernel):
To define the mth symmetric Volterra kernel, the system has to be excited with an mth-tone harmonic input. ◊

The 1st Volterra kernel can be extracted with a single tone input; thus, with $m = 1$, $u(t) = e^{s_1 t}$, $\mathbf{x}(t) = \sum_{k_1=0}^{\infty} \mathbf{G}_{k_1}(k_1 s_1) e^{k_1 s_1 t}$ and substituting to the system Eq. (4.101),

$$\sum_{k_1=0}^{\infty} (k_1 s_1 \mathbf{I} - \mathbf{A}) \mathbf{G}_{k_1}(k_1 s_1) e^{k_1 s_1 t} = \mathbf{B} e^{s_1 t} + \mathbf{N} \sum_{k_1=0}^{\infty} \mathbf{G}_{k_1}(k_1 s_1) e^{k_1 s_1 t} e^{s_1 t}, \tag{4.68}$$

the first kernel $\mathbf{G}_1$ is computed by setting $k_1 = 1$ and collecting the same exponential powers

$$(s_1 \mathbf{I} - \mathbf{A}) \mathbf{G}_1(s_1) = \mathbf{B} \Rightarrow \mathbf{G}_1(s_1) = (s_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{B}. \tag{4.69}$$

From the structure of the 1st kernel, it is evident that it involves all the operators a linear system contains. When good observables are known, a vector $\mathbf{C}$ with left multiplication leads to the classical input-output transfer function $H(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$. Although continuous bilinear systems can merit identification from input-output data, we want to relax the necessary reachability-observability assumptions in [104]. Therefore, we consider state snapshot data and the input-to-state transfer functions $\mathbf{G}_m$.

The 2nd Volterra kernel is with $m = 2$, $u(t) = e^{s_1 t} + e^{s_2 t}$, the steady state solution is $\mathbf{x}(t) = \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \mathbf{G}_{k_1+k_2}(k_1 s_1, k_2 s_2) e^{(k_1 s_1 + k_2 s_2)t}$, and substituting in the differential equation of the system, we get that

$$\begin{aligned} &\sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} ((k_1 s_1 + k_2 s_2)\mathbf{I} - \mathbf{A}) \mathbf{G}_{k_1+k_2}(k_1 s_1, k_2 s_2) e^{(k_1 s_1 + k_2 s_2)t} = \mathbf{B}(e^{s_1 t} + e^{s_2 t}) + \\ &+ \mathbf{N} \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \mathbf{G}_{k_1+k_2}(k_1 s_1, k_2 s_2) e^{(k_1 s_1 + k_2 s_2)t} (e^{s_1 t} + e^{s_2 t}), \end{aligned} \tag{4.70}$$

the $\mathbf{G}_2(s_1, s_2)$ after collecting the same exponential powers with $(s_1 + s_2)$ is

$$\begin{aligned} &((s_1 + s_2)\mathbf{I} - \mathbf{A}) \mathbf{G}_2(s_1, s_2) = \mathbf{N}(\mathbf{G}_1(s_1) + \mathbf{G}_1(s_2)) \Rightarrow \\ &\mathbf{G}_2(s_1, s_2) = ((s_1 + s_2)\mathbf{I} - \mathbf{A})^{-1} \mathbf{N}(\mathbf{G}_1(s_1) + \mathbf{G}_1(s_2)). \end{aligned} \tag{4.71}$$

Applying an mth ($m \geq 2$) tone harmonic input excitation to the bilinear system, the general mth order bilinear symmetric input to state Volterra kernel (symmetric transfer function) can be derived.

$$\mathbf{G}_m(s_1, \ldots, s_m) = ((s_1 + \cdots + s_m)\mathbf{I} - \mathbf{A})^{-1} \mathbf{N} \mathbf{R}_m(s_1, s_2, \ldots, s_m) \tag{4.72}$$

where $\mathbf{R}_m(s_1, s_2, \ldots, s_m)$ contains recursively the kernels $\mathbf{G}_1$, $\mathbf{G}_2, \ldots,$ $\mathbf{G}_{m-1}$.

As it is evident, Eq. (4.69) and Eq. (4.72) contain all the operators from the bilinear system ODE in Eq. (4.65). Towards the identification of these operators ($\mathbf{A}$, $\mathbf{N}$, $\mathbf{B}$), we start with the first kernel,

$$\begin{aligned}
&\mathbf{G}_1(s_1) = (s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} \Rightarrow (s_1\mathbf{I}_n - \mathbf{A})\mathbf{G}_1(s_1) = \mathbf{B} \Rightarrow\\
&\mathbf{A}\mathbf{G}_1(s_1) + \mathbf{B} = s_1\mathbf{G}_1(s_1) \Rightarrow \mathbf{I}_n\mathbf{A}\mathbf{G}_1(s_1) + \mathbf{B} = s_1\mathbf{G}_1(s_1) \Rightarrow\\
&(\mathbf{I}_n \otimes \mathbf{G}_1^T(s_1))vec(\mathbf{A}) + \mathbf{B} = s_1\mathbf{G}_1(s_1) \Rightarrow\\
&\left[\begin{array}{cc} \mathbf{I}_n \otimes \mathbf{G}_1^T(s_1) & \mathbf{I}_n \end{array}\right]\left[\begin{array}{c} \mathtt{vec}(\mathbf{A}) \\ \mathbf{B} \end{array}\right] = s_1\mathbf{G}_1(s_1)
\end{aligned} \tag{4.73}$$

To simplify the notation further, we introduce the resolvent $\mathbf{\Phi}_m(s_1, \dots, s_m)$ as

$$\mathbf{\Phi}_m(s_1, s_2, \dots, s_m) = ((s_1 + s_2 + \cdots + s_m)\mathbf{I}_n - \mathbf{A})^{-1} \in \mathbb{C}^{n\times n} \tag{4.74}$$

Now, Eq. (4.72) can be written as:

$$\begin{aligned}
&\mathbf{G}_m(s_1, s_2, \dots, s_m) = \mathbf{\Phi}_m(s_1, s_2, \dots, s_m)\mathbf{N}\mathbf{R}_m(s_1, s_2, \dots, s_m) \Rightarrow\\
&\mathbf{\Phi}_m^{-1}(s_1, s_2, \dots, s_m)\mathbf{G}_m(s_1, s_2, \dots, s_m) = \mathbf{N}\mathbf{R}_m(s_1, s_2, \dots, s_m) \Rightarrow\\
&((s_1 + s_2 + \cdots + s_m)\mathbf{I}_n - \mathbf{A})\mathbf{G}_m(s_1, s_2, \dots, s_m) = \mathbf{N}\mathbf{R}_m(s_1, s_2, \dots, s_m) \Rightarrow\\
&(s_1 + s_2 + \cdots + s_m)\mathbf{G}_m(s_1, s_2, \dots, s_m) = \mathbf{A}\mathbf{G}_m(s_1, s_2, \dots, s_m) + \mathbf{N}\mathbf{R}_m(s_1, s_2, \dots, s_m),
\end{aligned} \tag{4.75}$$

where, by denoting $\mathbf{G}_m = \mathbf{G}_m(s_1, s_2, \dots, s_m)$, $\mathbf{R}_m = \mathbf{R}_m(s_1, s_2, \dots, s_m)$ and using the property of the Kronecker product $\mathbf{X}\mathbf{Q}\mathbf{Y} = (\mathbf{X} \otimes \mathbf{Y}^T)\mathtt{vec}(\mathbf{Q})$, we conclude to

$$(s_1 + s_2 + \cdots + s_m)\mathbf{G}_m = \left(\mathbf{I}_n \otimes \mathbf{G}_m^T\right)\mathtt{vec}(\mathbf{A}) + \left(\mathbf{I}_m \otimes \mathbf{R}_m^T\right)\mathtt{vec}(\mathbf{N}). \tag{4.76}$$

Finally, by coupling the above two least-squares problems Eqs. (4.73-4.76), we derive for the arbitrary m-tone input, the following m-level least square problem. The solution of the following least-squares problem results in bilinear operator identification.

$$\left[\begin{array}{c} s_1\mathbf{G}_1(s_1) \\ \hline (s_1 + s_2)\mathbf{G}_2(s_1, s_2) \\ \vdots \\ (s_1 + \cdots + s_m)\mathbf{G}_m(s_1, \dots, s_m) \end{array}\right] = \underbrace{\left[\begin{array}{cc|c} \mathbf{I}_{n,n} & \mathbf{I}_n \otimes \mathbf{G}_1^T & \mathbf{0}_{n,n^2} \\ \hline \mathbf{0}_{n,n} & \mathbf{I}_n \otimes \mathbf{G}_2^T & \mathbf{I}_n \otimes \mathbf{R}_2^T \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{n,n} & \mathbf{I}_n \otimes \mathbf{G}_m^T & \mathbf{I}_n \otimes \mathbf{R}_m^T \end{array}\right]}_{\mathbf{M}} \left[\begin{array}{c} vec(\mathbf{B}) \\ vec(\mathbf{A}) \\ \hline vec(\mathbf{N}) \end{array}\right]. \tag{4.77}$$

Remark 4.21 (Spectral bilinear identification from state access):
When the $rank(\mathbf{M})$ is full, the operators $(\mathbf{A}, \mathbf{N}, \mathbf{B})$ can be identified. The resulting bilinear system is input invariant. Moreover, the snapshot data allow SVD-based projection schemes to construct reduced models similar to the methods, e.g., POD operator inference (OpInf). ♢

We will illustrate this method for a more general class of nonlinear dynamical systems in Section 5.6. In this thesis, we will keep the discussion and implementation minimal for approaches that require state-access snapshot data as we focus mainly on input-output data. Thus, we continue this chapter of modeling discrete bilinear control systems from input-output data.

4.5. Modeling discrete bilinear control systems

This section focuses on identifying and reducing discrete bilinear systems from time-domain data. These approaches are data-driven, i.e., the data are given by sampled trajectories of the control input and the observed output. We present connections to the time-domain Loewner method and the eigensystem realization (ERA) algorithm. The main challenge for extending these classical methods for fitting bilinear systems is the accurate recovery of Markov parameters from the input and output measurements. Afterward, one can employ a realization algorithm similar to that of Isidori [95]. The eigensystem realization algorithm (ERA) from Juang and Papa [99] provides a data-driven system identification method that discovers discrete systems capable of approximating the continuous impulse response from input-output data. The discrete bilinear model induces from the continuous one by employing discretization techniques (Euler or higher order schemes) with $t_k = k\Delta t$ as

$$
\begin{aligned}
&\mathbf{\Sigma}_{\text{cont}} : \begin{cases} \dot{\mathbf{x}}_c(t) = \mathbf{A}_c\mathbf{x}_c(t) + \mathbf{N}_c\mathbf{x}_c(t)u(t) + \mathbf{B}_c u(t), \\ y_c(t) = \mathbf{C}_c\mathbf{x}_c(t) + \mathbf{D}_d u(t). \end{cases} \quad \overset{\Delta t \to 0}{\underset{z_k(k\Delta t)}{\Longleftrightarrow}} \\
&\mathbf{\Sigma}_{\text{disc}} : \begin{cases} \mathbf{z}_{k+1} = \mathbf{A}_d\mathbf{z}_k + \mathbf{N}_d\mathbf{z}_k u_k + \mathbf{B}_d u_k, \\ \quad y_k = \mathbf{C}_d\mathbf{z}_k + \mathbf{D}_d u_k. \end{cases}
\end{aligned}
\tag{4.78}
$$

The subscripts "c" and "d" are the continuous and discrete operators, respectively.

Carleman's bilinearization Section 2.4.1 allows the bilinear system in Eq. (4.78) to approximate general analytic nonlinear systems. In the case of time domain data, the Hankel framework is a particular case of the Loewner framework. Ho and Kalman in [92] have provided the classical eigensystem realization algorithm (ERA) for linear system identification. Extensions to the bilinear identification through the subspace method have been accomplished from various studies [54, 46, 119]. The bottleneck for subspace methods is the accurate estimation of the state where the Kalman filter is not optimal and the dimensionality, as in the linear case. Isidori has provided theoretical results for input-output minimal bilinear realization [95]. By combining these concepts, we offer a new data-driven bilinear identification framework for time-domain data.

The nonlinear frameworks we developed can be used in real engineering applications and provide low-order nonlinear models directly from measurements capable of capturing the underlying nonlinearities and improving the performance of linear fits. Finally, these low-order nonlinear (bilinear) models allow the optimization of the engineering process via robust simulations and control design.

The authors in [54] have generalized the linear subspace identification theory to an analog theory for the subspace identification of bilinear systems. This approach is subject to white inputs that are mutually independent. The authors provided the link between the bilinear subspace method and the Kalman filter in the same work. The application concerns engineers' applications. In [54], the authors explain physical bilinear processes. For example, in most chemical processes, the controls are flow rates Eq. (4.79). From the first principles (mass and heat balances), we know these will appear in the system equations as products with the state variables (typical temperatures or concentrations). Therefore the bilinear equation for (continuously stirred tank reactors, distillation columns, etc.) can be written accordingly:

$$M\dot{\mathbf{x}} = \underbrace{\sum_i \mathbf{q}_i \mathbf{x}_i}_{\text{incoming flow}} - \underbrace{\sum_m \mathbf{q}_m \mathbf{x}_m}_{\text{outgoing flow}}, \ \mathbf{q}(\text{inputs}),\ \mathbf{x}(\text{state concentration}) \tag{4.79}$$

In [54], the MIMO bilinear system of the following type is considered:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{N}\mathbf{u}_k \otimes \mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \boldsymbol{\omega}_k, \\ \quad \mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \upsilon_k. \end{cases} \tag{4.80}$$

Assumption 1: The inputs are assumed to be observed and white. Moreover, they should be mutually independent and independent of the measurement noise υ_k and the process noise $\boldsymbol{\omega}_k$. The covariance matrix of these sequences is

$$\mathbf{E}\left[\begin{pmatrix} \mathbf{u}_p \\ \boldsymbol{\omega}_p \\ \upsilon_p \end{pmatrix} \begin{pmatrix} \mathbf{u}_p^T & \boldsymbol{\omega}_p^T & \upsilon_p^T \end{pmatrix}\right] = \begin{bmatrix} \mathbf{I}_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{S} \\ \mathbf{0} & \mathbf{S}^T & \mathbf{R} \end{bmatrix} \delta_{pq} \geq 0. \tag{4.81}$$

Problem statement: Given measurements of the inputs $\mathbf{u}_k$ and the outputs $\mathbf{y}_k$ of the unknown bilinear system, determine the system matrices $(\mathbf{A}, \mathbf{N}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ as an estimate of the noise covariance matrices $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{S}$. Prediction error methods [49, 3] aim to find the exact solution. Subspace identification techniques have advantages: they provide a state-space representation of the unknown system by projecting the row spaces of matrices constructed only from input and output data. PEM and subspace methods can be generalized to MIMO systems and can be direct.

The authors in [44] (Chen et al. 2000) succeeded in relaxing the hypothesis in [54] to the case where the measured input is not restricted to be white. Moreover, the results are not biased estimates of the system as in [54]. Further, in [49], the authors determine bilinear Markov parameters using unit impulses at the discrete-time points $k = 0$ and $k = i_1$, etc. Thus, after solving the linear system, he computes bilinear Markov parameters and constructs the bilinear Hankel that leads to reduced discrete bilinear systems that finally can be transformed to continuous bilinear via the following transformation:

Theorem 4.22 (Dorissen 1990 [49]):
A sampled continuous-time bilinear system with sampling rate T answers with Markov parameters of an equivalent discrete-time bilinear model as:

$$\begin{aligned} \mathbf{A}_d &= e^{\mathbf{A}_c T}, \\ \mathbf{N}_d &= e^{(\mathbf{A}_c + \mathbf{N}_c)T} - e^{\mathbf{A}_c T}, \\ \mathbf{B}_d &= \int_0^T e^{(\mathbf{A}_c + \mathbf{N}_c)\tau} \mathbf{B}_c d\tau, \\ \mathbf{C}_d &= \mathbf{C}_c, \end{aligned} \tag{4.82}$$

if it is excited by a sequence of unit discrete impulses

$$u(k) = \begin{cases} 1 & \forall\ k = i_1, i_2, \dots \text{ and } 0 \leq i_1 < i_2 < \dots \\ 0 & \text{otherwise} \end{cases}. \tag{4.83}$$

◊

Correspondingly, from the realization $(\mathbf{A}_d, \mathbf{N}_d, \mathbf{B}_d, \mathbf{C}_d)$ of the evaluated Markov parameters, the continuous time BLs can be computed as above. The authors in [119] can construct bilinear systems with white noise inputs based on an iterative deterministic-stochastic subspace approach. They have the same assumption as [54] on the random processes. To develop their bilinear system identification algorithm, a characterization of the bilinear portion of the state equation $\mathbf{z}(t) = \mathbf{u}(t) \otimes \mathbf{x}(t)$, as a second-order stationary white noise process is required.

The author in [98] uses the linear model properties of the bilinear system when subjected to constant input. Constant inputs can transform the bilinear model into an equivalent linear model as in [73]. In [98], identification can be achieved by identifying the matrices $\mathbf{A}, \mathbf{C}, \mathbf{D}$ and, secondly, the matrices $\mathbf{B}, \mathbf{N}$.

The model depends on the bilinear minimality and stays consistent with the linear model. Although the complexity grows exponentially, which is common for nonlinear identification frameworks (e.g., subspace identification method for bilinear), the repetitive process can be implemented in parallel as all the experiments can be obtained independently. The short simulation time for each experiment is enough to exploit the information of the nonlinear model, as the choice of white noise excitation inputs can activate enough modes (persistent excitation) to achieve identification. Moreover, constructing the inputs can give the matrix $\mathbf{U}$ complete rank for the deterministic case. As with bilinear systems, white inputs with small magnitudes can overload with big-magnitude inputs or constant inputs, and the result can be unstable.

4.5.1. The Moebius transform for the discretization of dynamical systems

The following exact transformation describes the mapping between the continuous Laplace s-domain and the discrete z-domain

$$z = e^{s\tau}, \ \tau\text{: sampling rate} \tag{4.84}$$

As the exact transformation will produce non-rational transfer functions, we approximate it with the Taylor series by splitting first the exponential as (Tustin's method)

$$z = e^{s\tau} = \frac{e^{s\tau/2}}{e^{-s\tau/2}} = \frac{1 + \frac{s\tau}{2} + \frac{(s\tau)^2 + \cdots}{4 \cdot 2!}}{1 - \frac{s\tau}{2} + \frac{(s\tau)^2 + \cdots}{4 \cdot 2!}} \approx \frac{1 + \frac{s\tau}{2}}{1 - \frac{s\tau}{2}} \tag{4.85}$$

where the bilinear transform and the inverse map are precisely the following:

$$z = \frac{1 + \frac{\tau}{2}s}{1 - \frac{\tau}{2}s} \Leftrightarrow s = \frac{2}{\tau}\frac{z-1}{z+1} \tag{4.86}$$

Let's first check the influence that this approximation (Moebius transformation) has on the first transfer function

$$H_1(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \; s \in \mathbb{C} \tag{4.87}$$

and by substituting the $s \to \frac{2}{\tau}\frac{z-1}{z+1}$ we have:

$$G_1(z) = H_1\left(\frac{2}{\tau}\frac{z-1}{z+1}\right) = \mathbf{C}\left(\frac{2}{\tau}\frac{z-1}{z+1}\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D} \tag{4.88}$$

a useful identity, $x(x\mathbf{I} + \mathbf{A})^{-1} = \mathbf{I} - \mathbf{A}(x\mathbf{I} + \mathbf{A})^{-1}$. To make it more general, we assume the general bilinear transform as the following mapping

$$s \to \frac{az + b}{cz + d}. \tag{4.89}$$

We compute

$$\begin{aligned}
G_1(z) &= H_1(s(z)) = H_1\left(\frac{az+b}{cz+d}\right) = \mathbf{C}\left(\frac{az+b}{cz+d}\mathbf{E} - \mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D}\\
&= (cz+d)\mathbf{C}\left((az+b)\mathbf{E} - (cz+d)\mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D}\\
&= (cz+d)\mathbf{C}\left(z\underbrace{(a\mathbf{E} - c\mathbf{A})}_{\mathbf{M}_1} + \underbrace{(b\mathbf{E} - d\mathbf{A})}_{\mathbf{M}_0}\right)^{-1}\mathbf{B} + \mathbf{D}\\
&= c\mathbf{C}z\left(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0\right)^{-1}\mathbf{M}_1^{-1}\mathbf{B} + d\mathbf{C}\left(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0\right)^{-1}\mathbf{M}_1^{-1}\mathbf{B} + \mathbf{D}\\
&= c\mathbf{C}\left[\mathbf{I} - \mathbf{M}_1^{-1}\mathbf{M}_0\left(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0\right)^{-1}\right]\mathbf{M}_1^{-1}\mathbf{B} + d\mathbf{C}\left(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0\right)^{-1}\mathbf{M}_1^{-1}\mathbf{B} + \mathbf{D}\\
&= c\mathbf{C}\mathbf{M}_1^{-1}\mathbf{B} + (d\mathbf{C} - c\mathbf{C}\mathbf{M}_1^{-1}\mathbf{M}_0)\left(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0\right)^{-1}\mathbf{M}_1^{-1}\mathbf{B} + \mathbf{D} \Leftrightarrow\\
G_1(z) &= \underbrace{(d\mathbf{C} - c\mathbf{C}\mathbf{M}_1^{-1}\mathbf{M}_0)}_{\mathbf{C}_d}\left(z\mathbf{I} - \underbrace{(-\mathbf{M}_1^{-1}\mathbf{M}_0)}_{\mathbf{A}_d}\right)^{-1}\underbrace{\mathbf{M}_1^{-1}\mathbf{B}}_{\mathbf{B}_d} + \underbrace{c\mathbf{C}\mathbf{M}_1^{-1}\mathbf{B} + \mathbf{D}}_{\mathbf{D}_d}
\end{aligned}$$

Expressing the discrete operators to the continuous operators, we can write for the general bilinear (Moebius) transformation the following:

$$\begin{aligned}
\mathbf{A}_d &= -\mathbf{M}_1^{-1}\mathbf{M}_0 = -(a\mathbf{E} - c\mathbf{A})^{-1}(b\mathbf{E} - d\mathbf{A})\\
\mathbf{B}_d &= \mathbf{M}_1^{-1}\mathbf{B} = (a\mathbf{E} - c\mathbf{A})^{-1}\mathbf{B}\\
\mathbf{C}_d &= d\mathbf{C} - c\mathbf{C}\mathbf{M}_1^{-1}\mathbf{M}_0) = d\mathbf{C} - c\mathbf{C}(a\mathbf{E} - c\mathbf{A})^{-1}(b\mathbf{E} - d\mathbf{A})\\
\mathbf{D}_d &= c\mathbf{C}\mathbf{M}_1^{-1}\mathbf{B} + \mathbf{D} = c\mathbf{C}(a\mathbf{E} - c\mathbf{A})^{-1}\mathbf{B} + \mathbf{D}
\end{aligned} \tag{4.90}$$

Therefore, the connection between the continuous and the discrete operators with the general bilinear transform approximation and for the linear system remains:

$$\begin{aligned}
\mathbf{A}_d &= -(a\mathbf{E} - c\mathbf{A})^{-1}(b\mathbf{E} - d\mathbf{A})\\
\mathbf{B}_d &= (a\mathbf{E} - c\mathbf{A})^{-1}\mathbf{B}\\
\mathbf{C}_d &= d\mathbf{C} - c\mathbf{C}(a\mathbf{E} - c\mathbf{A})^{-1}(b\mathbf{E} - d\mathbf{A})\\
\mathbf{D}_d &= c\mathbf{C}(a\mathbf{E} - c\mathbf{A})^{-1}\mathbf{B} + \mathbf{D}
\end{aligned} \tag{4.91}$$

It is interesting to investigate how these matrices are affected under Euler schemes.

- The Forward Euler (FE). The derivative is approximated by: $\frac{df(t)}{dt} = \frac{f(t+\tau)-f(t)}{\tau}$ where in the Laplace domain, we have $sf(s) = \frac{zf(s)-f(s)}{\tau} \rightarrow \tau s = z-1 \Leftrightarrow z = 1+\tau s$. Thus, for the FE, the transformation is $s = \frac{z-1}{0\cdot z+\tau}$, and the parameters are FE $(a,b,c,d) = (1,-1,0,\tau)$. The discrete operators are:

$$\begin{aligned}
\mathbf{A}_d &= -(\mathbf{E})^{-1}(-\mathbf{E}-\tau\mathbf{A}) = (\mathbf{I}+\tau\mathbf{E}^{-1}\mathbf{A}) \;\; \text{stability: } \|\cdot\| \leq 1\\
\mathbf{B}_d &= \mathbf{E}^{-1}\mathbf{B}\\
\mathbf{C}_d &= \tau\mathbf{C}\\
\mathbf{D}_d &= \mathbf{D}.
\end{aligned} \tag{4.92}$$

- The Backward Euler (BE). The derivative is approximated by: $\frac{df(t)}{dt} = \frac{f(t)-f(t-\tau)}{\tau}$ where in the Laplace domain, we have $sf(s) = \frac{f(s)-f(s)z^{-1}}{\tau} \rightarrow \tau s = 1-z^{-1} \Leftrightarrow z = \frac{1}{1-\tau s}$. Thus, for the BE, the transformation is $s = \frac{z-1}{\tau z+0}$, and the parameters are BE $(a,b,c,d) = (1,-1,\tau,0)$. The discrete operators are:

$$\begin{aligned}
\mathbf{A}_d &= -(\mathbf{E}-\tau\mathbf{A})^{-1}(-\mathbf{E}) = (\mathbf{I}-\tau\mathbf{E}^{-1}\mathbf{A})^{-1} \;\; \text{stability: } \|\cdot\| \leq 1\\
\mathbf{B}_d &= (\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{B} = (\mathbf{I}-\tau\mathbf{E}^{-1}\mathbf{A})^{-1}\mathbf{E}^{-1}\mathbf{B}\\
\mathbf{C}_d &= -\tau\mathbf{C}(\mathbf{E}-\tau\mathbf{A})^{-1}(-\mathbf{E}) = \tau\mathbf{C}(\mathbf{I}-\tau\mathbf{A})^{-1}\\
\mathbf{D}_d &= \tau\mathbf{C}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{B}+\mathbf{D} = \tau\mathbf{C}(\mathbf{I}-\tau\mathbf{E}^{-1}\mathbf{A})^{-1}\mathbf{E}^{-1}\mathbf{B}+\mathbf{D}.
\end{aligned} \tag{4.93}$$

Remark 4.23 (Structure preservation for LTI systems with BE):
The Backward Euler preserves the structure of the original linear system as it contains the same operators. These are $(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$. The above result proves that the discrete and the continuous models are equivalent for sampling time that tends to zero. Moreover, if a $\mathbf{D}$ term has been assumed zero for the continuous model, the discrete contains a non-zero $\mathbf{D}$ term. ♢

4.5.2. The generalized bilinear transform and discretization schemes

There are several ways of approximating a continuous model with a discrete one of the same structure. Most prominent and well-studied for comparing the derived are the Euler schemes (e.g., the backward or the forward Euler), where the simplicity of applying these has shaped the class of discretization methods. Despite the numerical stability issues frequently arising in compliance with spatial discretization, such as 1st order schemes, they have treated many applications and performed well. Central or 2nd order schemes like the Cranck-Nicolson or Wilson θ methods have also been applied to improve the discretization error. There are general polynomial schemes for improving the derivative approximation, e.g., higher-order schemes (Range Kutta).

Following [53], the discretization of a continuous bilinear system can be exact when the bilinear system can be transformed to the phase-canonical form. That can be possible when the linear subsystem has a full-rank controllability matrix. This is very rare in real applications, which immediately makes the subclass of these bilinear systems quite restrictive.

To illustrate the effect of discretization from the frequency domain, we introduce the bilinear transform and apply it to the 2nd generalized frequency response function for the bilinear case. The 2nd generalized regular kernel for the bilinear systems is

$$H_2(s_1, s_2) = \mathbf{C}(s_2\mathbf{I} - \mathbf{A})^{-1}\mathbf{N}(s_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \tag{4.94}$$

We apply the general bilinear transform for each argument as before $s = \frac{az+b}{cz+d}$. Therefore, we are trying to create an equivalent discrete bilinear model with a transfer function as

$$\begin{aligned} G_2(z_1, z_2) = H_2(s_1(z_1), s_2(z_2)) &= H_2\left(\frac{az_1+b}{cz_1+d}, \frac{az_2+b}{cz_2+d}\right) \\ &= \mathbf{C}\left(\frac{az_1+b}{cz_1+d}\mathbf{E} - \mathbf{A}\right)^{-1}\mathbf{N}\left(\frac{az_2+b}{cz_2+d}\mathbf{E} - \mathbf{A}\right)^{-1}\mathbf{B}. \end{aligned} \tag{4.95}$$

We isolate the resolvent to perform the following computations:

$$\begin{aligned} \left(\frac{az+b}{cz+d}\mathbf{E} - \mathbf{A}\right)^{-1} &= (cz+d)\left((az+b)\mathbf{E} - (cz+d)\mathbf{A}\right)^{-1} \\ &= (cz+d)\left(z\underbrace{(a\mathbf{E} - c\mathbf{A})}_{\mathbf{M}_1} + \underbrace{(b\mathbf{E} - d\mathbf{A})}_{\mathbf{M}_0}\right)^{-1} \\ &= (cz+d)(z\mathbf{M}_1 + \mathbf{M}_0)^{-1} = (cz+d)\left[\mathbf{M}_1\left(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0\right)\right]^{-1} \\ &= (cz+d)(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0)^{-1}\mathbf{M}_1^{-1} \\ &= c\underbrace{z(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0)^{-1}}_{\text{identity}}\mathbf{M}_1^{-1} + d(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0)^{-1}\mathbf{M}_1^{-1} \\ &= c\left[\mathbf{I} - \mathbf{M}_1^{-1}\mathbf{M}_0(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0)^{-1}\right]\mathbf{M}_1^{-1} + d(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0)^{-1}\mathbf{M}_1^{-1} \\ &= c\mathbf{M}_1^{-1} - c\mathbf{M}_1^{-1}\mathbf{M}_0(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0)^{-1}\mathbf{M}_1^{-1} + d(z\mathbf{I} + \mathbf{M}_1^{-1}\mathbf{M}_0)^{-1}\mathbf{M}_1^{-1} \\ &= c\mathbf{M}_1^{-1} + (d\mathbf{I} - c\mathbf{M}_1^{-1}\mathbf{M}_0)(z\mathbf{I} + \underbrace{\mathbf{M}_1^{-1}\mathbf{M}_0}_{-\mathbf{A}_d})^{-1}\mathbf{M}_1^{-1} \\ &= c\mathbf{M}_1^{-1} + \underbrace{(d\mathbf{I} + c\mathbf{A}_d)}_{\mathbf{H}}(z\mathbf{I} - \mathbf{A}_d)^{-1}\underbrace{\mathbf{M}_1^{-1}}_{\mathbf{G}} \\ &= c\mathbf{G} + \mathbf{H}(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{G}. \end{aligned}$$

Back to $G_2(z_1, z_2)$, we have

$$\begin{aligned} G_2(z_1, z_2) &= \mathbf{C}\left(c\mathbf{G} + \mathbf{H}(z_2\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{G}\right)\mathbf{N}\left(c\mathbf{G} + \mathbf{H}(z_1\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{G}\right)\mathbf{B} \\ &= c^2\mathbf{CGNGB} + \\ &+ c\mathbf{CGNH}(z_1\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{GB} + c\mathbf{CH}(z_2\mathbf{I} - A_d)^{-1}\mathbf{GNGB} + \\ &\underbrace{\mathbf{CH}(z_2\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{GNH}(z_1\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{GB}}_{\approx G_2(z_1, s_2)\text{ bilinear form}}. \end{aligned} \tag{4.96}$$

Here, we write all the above moments w.r.t the continuous operators of the bilinear system.

$$\begin{aligned} c^2\mathbf{CGNGB} &= c^2\mathbf{C}(a\mathbf{E}-c\mathbf{A})^{-1}\mathbf{N}(a\mathbf{E}-c\mathbf{A})^{-1}\mathbf{B} \\ &\overset{\text{BE}}{=} \tau^2\mathbf{C}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{N}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{B} \overset{\tau\to 0}{\to} 0 \end{aligned} \tag{4.97}$$

the next one,

$$\begin{aligned} &c\mathbf{CGNH}(z_1\mathbf{I}-\mathbf{A}_d)^{-1}\mathbf{GB} = c\mathbf{C}(a\mathbf{E}-c\mathbf{A})^{-1}\mathbf{N}(d\mathbf{I}+c\mathbf{A}_d)(z_1\mathbf{I}-\mathbf{A}_d)^{-1}(a\mathbf{E}-c\mathbf{A})^{-1}\mathbf{B} \\ &\overset{\text{BE}}{=} -\tau\mathbf{C}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{N}\tau(\mathbf{E}-\tau\mathbf{A})^{-1}(-\mathbf{E})\left(z_1\mathbf{I}+(\mathbf{E}-\tau\mathbf{A})^{-1}(-\mathbf{E})\right)^{-1}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{B} \\ &= \tau^2\mathbf{C}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{N}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{E}\left(\frac{1}{1-s_1\tau}\mathbf{I}+(\mathbf{E}-\tau\mathbf{A})^{-1}(-\mathbf{E})\right)^{-1}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{B} \\ &= \mathbf{C}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{N}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{E}\underbrace{\left(\frac{1}{(1-s_1\tau)\tau^2}\mathbf{I}+\frac{1}{\tau^2}(\mathbf{E}-\tau\mathbf{A})^{-1}(-\mathbf{E})\right)^{-1}}_{\mathbf{\Phi}(\tau)}(\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{B} \end{aligned}$$

where we have the following two limits:

$$\lim_{\tau\to 0}\frac{\frac{1}{1-s_1\tau}}{\tau^2} = \frac{1}{0^+} = +\infty, \quad \lim_{\tau\to 0}\frac{(-\mathbf{E}-\tau\mathbf{A})^{-1}\mathbf{E}}{\tau^2} = \frac{-1}{0^+} = -\infty \tag{4.98}$$

since the indeterminacy remains as $(+\infty)+(-\infty)=?$, therefore, we need to combine.

$$\begin{aligned} \lim_{\tau\to 0}\mathbf{\Phi}^{-1}(\tau) &= \lim_{\tau\to 0}\frac{\mathbf{I}-(1-s_1\tau)(\mathbf{I}-\tau\mathbf{E}^{-1}\mathbf{A})^{-1}}{\tau^2} \overset{0/0}{=} \\ \lim_{\tau\to 0}\mathbf{\Phi}^{-1}(\tau) &= \lim_{\tau\to 0}\frac{s_1(\mathbf{I}-\tau\mathbf{E}^{-1}\mathbf{A})^{-1}-(1-s_1\tau)(\mathbf{E}^{-1}\mathbf{A})(\mathbf{I}-\tau\mathbf{E}^{-1}\mathbf{A})^{-2}}{2\tau} = \frac{s_1\mathbf{I}-\mathbf{E}^{-1}\mathbf{A}}{2\cdot 0^+} = \pm\infty\cdot\mathbf{1} \\ \lim_{\tau\to 0}\mathbf{\Phi}(\tau) &= (\pm\infty\cdot\mathbf{1})^{-1} = \mathbf{0}. \end{aligned} \tag{4.99}$$

Finally, the remaining quantities preserve the bilinear structure with the BE scheme only asymptotically when $\tau \to 0$ ($G_2(z_1,z_2)$ ≈bilinear$+O(\tau)$(other terms)). Thus, the following equivalences between the continuous and discrete operators are presented next:

$$\begin{aligned} \mathbf{E}_d &= \mathbf{I}, \\ \mathbf{A}_d &= (\mathbf{I}-\tau\mathbf{A})^{-1}, \\ \mathbf{N}_d &= \tau(\mathbf{I}-\tau\mathbf{A})^{-1}\mathbf{N}(\mathbf{I}-\tau\mathbf{A})^{-1}, \\ \mathbf{B}_d &= (\mathbf{I}-\tau\mathbf{A})^{-1}\mathbf{B}, \\ \mathbf{C}_d &= \tau\mathbf{C}(\mathbf{I}-\tau\mathbf{A})^{-1}, \\ \mathbf{D}_d &= \mathbf{D}+\tau\mathbf{C}(\mathbf{I}-\tau\mathbf{A})^{-1}\mathbf{B}. \end{aligned} \tag{4.100}$$

Remark 4.24 (Discretization of bilinear control systems):
As detailed in the above analysis, discretization of the nonlinear systems, even in the simplest case, that of bilinear, is not easy. In particular, besides the forward Euler that indeed preserves the structure, the rest of the discretization schemes do not preserve the structure inserting an additional error. Exact discretization of bilinear systems can only be obtained with very restrictive assumptions [49]. ◊

4.6. Discrete bilinear realization

Ho and Kalman have set the mathematical foundations for achieving minimal system realization in linear systems. They have provided the eigensystem realization algorithm (ERA) that computationally allows model construction for identification and reduction directly from data. In the nonlinear case and towards the exact scope of identifying nonlinear systems, Isidori in [95] has extended these results for the bilinear case. This section presents all the mathematical results for constructing minimal bilinear realization from nonlinear input-output maps. At the same time, the algorithm for achieving this goal comes along by circumventing some of the computational challenges.

This approach differs from the classical subspace identification approaches, but as it appeared in [49], the computational challenges and some assumptions remain the same. Next is the single-input single-output (SISO) continuous bilinear system of state dimension n.

$$\mathbf{\Sigma}_{\text{cont}}:\begin{cases}\dot{\mathbf{x}}_c(t)=\mathbf{A}_c\mathbf{x}_c(t)+\mathbf{N}_c\mathbf{x}_c(t)u(t)+\mathbf{B}_cu(t),\\ y_c(t)=\mathbf{C}_c\mathbf{x}_c(t)+\mathbf{D}_cu(t),\ \mathbf{x}_c(0)=\mathbf{x}_0=\mathbf{0},\end{cases}\tag{4.101}$$

where with sampling time dt, the discrete model at $0<dt<2dt<\cdots<kdt$, with $\mathbf{x}_c(kdt)=\mathbf{x}_k$ and $u(kdt)=u_k$ for $k=0,\ldots,m-1$ are defined as:

$$\mathbf{\Sigma}_{\text{disc}}:\begin{cases}\mathbf{x}_{k+1}=\mathbf{A}\mathbf{x}_k+\mathbf{N}\mathbf{x}_ku_k+\mathbf{B}u_k,\\ y_k=\mathbf{C}\mathbf{x}_k+\mathbf{D}u_k,\ \mathbf{x}_0=\mathbf{0}.\end{cases}\tag{4.102}$$

The system in Eq. (4.102) has state dimension n, so, $\mathbf{x}\in\mathbb{R}^n$ and the operators have dimensions $\mathbf{A},\mathbf{N}\in\mathbb{R}^{n\times n}$, $\mathbf{B},\mathbf{C}^T\in\mathbb{R}^n$, and $\mathbf{D}\in\mathbb{R}$. We assume that the system starts with homogeneous initial conditions. Although the $\mathbf{D}$ term is manageable, here, for simplicity, it is considered zero.

The following step is potentially analogous to the linear case whereby, by exciting the system with the unit impulse, the output (impulse response) consists of the Markov parameters that form the linear Hankel matrix and lead to identification. The bilinear case is more complex in extracting the bilinear Markov parameters and constructing the bilinear Hankel matrix. Moreover, as it is normal for the existing nonlinear identification frameworks (e.g., subspace methods), the complexity class is exponential, and the "curse of dimensionality" needs careful treatment, and maybe some possible solutions can be stated in what follows.

Definition 4.25 (The reachability matrix):
The matrix $\mathcal{P}_n=\begin{bmatrix}\mathbf{P}_1 & \cdots & \mathbf{P}_n\end{bmatrix}$ is defined recursively from the following relation:

$$\mathbf{P}_j=\begin{bmatrix}\mathbf{A}\mathbf{P}_{j-1} & \mathbf{N}\mathbf{P}_{j-1}\end{bmatrix},\ j=2,\ldots n,\ \mathbf{P}_1=\mathbf{B}.$$

Then the state space of the bilinear system is spanned by the states reachable from the origin if and only if $\texttt{rank}(\mathcal{P}_n)=n$.

Definition 4.26 (The observability matrix):
The matrix $\mathcal{Q}_n^T=\begin{bmatrix}\mathbf{Q}_1 & \cdots & \mathbf{Q}_n\end{bmatrix}^T$ is defined recursively from the following relation:

$$\mathbf{Q}_j^T=\begin{bmatrix}\mathbf{Q}_{j-1}\mathbf{A} & \mathbf{Q}_{j-1}\mathbf{N}\end{bmatrix}^T,\ j=2,\ldots n,\ \mathbf{Q}_1=\mathbf{C}.$$

Then the state space of the bilinear system is observable if and only if $\texttt{rank}(\mathcal{Q}_n) = n$. The following input definition allows a concise representation of the input-output relation.

Definition 4.27 (Input sequence):
The block structure of the input sequence through the time steps is given as

$$\mathbf{u}_j(h) = \begin{bmatrix} \mathbf{u}_{j-1}(h) \\ \mathbf{u}_{j-1}(h)u(h+j-1) \end{bmatrix}, \; j = 2, \dots, \; \mathbf{u}_1(h) = u(h).$$

Let $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_j, \dots\}$ be an infinite sequence of row vectors, in which $\mathbf{w}_j \in \mathbb{R}^{1\times 2^{j-1}}$ and is defined recursively as follows $\mathbf{w}_j = \mathbf{C}\mathbf{P}_j, \; j = 1, 2, \dots;$

The state response of system Eq. (4.102) from the state $\mathbf{x}_0 = \mathbf{0}$ at time $k = 0$, under a given input function can be expressed as:

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{B}u_0 \triangleq \mathbf{P}_1\mathbf{u}_1(0), \\
\mathbf{x}_2 &= \mathbf{A}\mathbf{P}_1\mathbf{u}_1(0) + \mathbf{N}\mathbf{P}_1\mathbf{u}_1(0)u(1) + \mathbf{B}u(1) \triangleq \mathbf{P}_2\mathbf{u}_2(0) + \mathbf{P}_1\mathbf{u}_1(1), \\
\mathbf{x}_3 &= \mathbf{A}\left(\mathbf{P}_2\mathbf{u}_2(0) + \mathbf{P}_1\mathbf{u}_1(1)\right) + \mathbf{N}\left(\mathbf{P}_2\mathbf{u}_2(0) + \mathbf{P}_1\mathbf{u}_1(1)\right)u(2) + \mathbf{B}u(2) \\
&\triangleq \mathbf{P}_2\mathbf{u}_3(0) + \mathbf{P}_2\mathbf{u}_2(1) + \mathbf{P}_1\mathbf{u}_1(2), \\
&\vdots \\
\mathbf{x}_k &= \sum_{j=1}^{k} \mathbf{P}_j\mathbf{u}_j(k-j), \; k = 1, 2, \dots;
\end{aligned} \tag{4.103}$$

Finally, the zero-state input-output map of system Eq. (4.102) after multiplication with the vector $\mathbf{C}$ from the left can be written as:

$$y_k = \sum_{j=1}^{k} \mathbf{w}_j\mathbf{u}_j(k-j), \; k = 1, 2, \dots; \tag{4.104}$$

The input-output map in Eq. (4.104) indicates that the state-space bilinear model is equivalent to the infinite sequence of row vectors $\{\mathbf{w}_j\}$. Therefore, the *complete bilinear realization problem* can be stated as follows: *given an infinite sequence of row vectors* $\{\mathbf{w}_j\}$*, find a quadruplet of matrices* $(\mathbf{A}, \mathbf{N}, \mathbf{B}, \mathbf{C})$ *such that Eq. (4.104) holds for all* $j = 1, 2, \dots;$ If j is finite ($j \leq M < \infty$), then, the identification restricted to the *partial bilinear realization* problem by matching up to M.

The $\{\mathbf{w}_j\}$ vector sequence is important because it stays invariant regarding the input-output data. In the linear case, these invariant quantities are called Markov parameters and can easily be extracted after an impulse excitation or by solving a linear least square problem. In the bilinear case, several simulations are needed with short lengths to recover the bilinear Markov parameters encoded in the $\{\mathbf{w}_j\}$ vector sequence. Once the bilinear Markov parameters are recovered, the solution of the bilinear realization problem can be computed with a simple algorithm that extends the linear ERA.

4.6.1. The bilinear Markov parameters

First, we must compute the bilinear Markov parameters to achieve bilinear realization. The bilinear Markov parameters are encoded in the $\{\mathbf{w}_j\}$-vectors for $j = 1, \ldots, n$, that appear in Eq. (4.104). These parameters play the role of the invariant quantities for the bilinear system, and the connection between the input and the output, although linear, the underlined linear least squares system is under-determined.

$$\begin{aligned} y_1 &= \mathbf{w}_1\mathbf{u}_1(0), \\ y_2 &= \mathbf{w}_1\mathbf{u}_1(1) + \mathbf{w}_2\mathbf{u}_2(0), \\ &\vdots \\ y_k &= \mathbf{w}_1\mathbf{u}_1(k-1) + \mathbf{w}_2\mathbf{u}_2(k-2) + \cdots + \mathbf{w}_k\mathbf{u}_k(0). \end{aligned} \tag{4.105}$$

Assembling the above equations appropriately into a matrix format, we can write

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} \mathbf{u}_1^T(0) & 0 & \cdots & 0 \\ \mathbf{u}_1^T(1) & \mathbf{u}_2^T(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_1^T(k-1) & \mathbf{u}_2^T(k-2) & \cdots & \mathbf{u}_k^T(0) \end{bmatrix}}_{\mathbf{U}} \cdot \underbrace{\begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_k^T \end{bmatrix}}_{\mathbf{W}}. \tag{4.106}$$

where the dimensions are: $\mathbf{Y} \in \mathbb{R}^{k\times 1}$, $\mathbf{U} \in \mathbb{R}^{k\times m}$, and $\mathbf{W} \in \mathbb{R}^{m\times 1}$. The integer m counts the bilinear Markov parameters activated at the time step k. This can be computed from the finite geometric series sum of the first k terms with growth 2. Thus, $m = \sum_{i=0}^{k} 2^i = 2^k - 1$.

As expected, the problem cannot be solved directly by applying only linear operations for the nonlinear case. In particular, the difficulty in treating this problem stems in Eq. (4.106), and the bilinear term causes the complexity to increase exponentially. Therefore, the least square problem filled out with k time steps will remain under-determined for $\forall k \in \{2, 3, \ldots\}$ as long as $2^k - 1$ bilinear Markov parameters are activated. Continuously, we must deal with a least square problem with k equations and $2^k - 1$ unknowns.

Solving an under-determined system is not impossible, but the solutions are infinite, and regularization schemes cannot easily lead to identification. Therefore, one way of uniquely identifying the bilinear Markov parameters and determining the $\mathbf{W}$ solution vector can be achieved by solving a coupled least squares system after applying several simulations to the original system that could be a black box or a practical engineering application (e.g., chemical reactor).

As the $\mathbf{W}$ vector encodes the bilinear Markov parameters and remains invariant concerning the input-output data, to determine $2^k - 1$ parameters, the column rank of the matrix $\mathbf{U}$ should be complete. That can be accomplished by augmenting rows underneath the matrix $\mathbf{U}$ until the new extended matrix $\hat{\mathbf{U}}$ will be square by concatenating several $\mathbf{U}$ matrices from the different simulations. To determine $2^k - 1$ parameters, we need 2^{k-1} simulations of the original system. That is precisely the expected bottleneck for nonlinear identification frameworks that deal with time-domain data. This result is reasonable from another point of view, which will be analyzed in more detail later. Still, here we can predict that the length of the Markov parameters should form a square Hankel matrix with winding, which is approximately the square root of this length. The coupled least squares problem is:

$$\underbrace{\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_d \end{bmatrix}}_{\hat{\mathbf{Y}}} = \underbrace{\begin{bmatrix} \cdots & \mathbf{U}_1 & \cdots \\ \cdots & \mathbf{U}_2 & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \mathbf{U}_d & \cdots \end{bmatrix}}_{\hat{\mathbf{U}}} \cdot \mathbf{W} \tag{4.107}$$

where $d = 2^{k-1}$. Since we repeat the simulation d times, and each time we get k equations, with the i^{th} simulation to be $\mathbf{Y}_i = \begin{bmatrix} y_1^{(i)} & y_2^{(i)} & \cdots & y_k^{(i)} \end{bmatrix}^T$ and accordingly for the $\mathbf{U}_i$, the real matrix $\hat{\mathbf{U}}$ has dimension $2^k \times (2^k - 1)$. After concatenating all the lower triangular matrices with full column rank, the $\hat{\mathbf{U}}$ matrix results. To certify that the $\hat{\mathbf{U}}$ will also have full column rank, one choice is to use white inputs (sampled from a Gaussian distribution) for the simulations. The use of white inputs is very common for system identification. Still, in that case, a careful choice of deterministic inputs can make the inversion exact and capable of recovering the bilinear Markov parameters. The least squares solution is:

$$\texttt{rank}(\hat{\mathbf{U}}) = 2^k - 1, \text{ the unique solution is: } \mathbf{W} = \hat{\mathbf{U}}^{-1}\hat{\mathbf{Y}} \in \mathbb{R}^{2^k-1}, \tag{4.108}$$

where the vector $\mathbf{W}$ contains the $2^k - 1$, bilinear Markov parameters. As we have computed the bilinear Markov parameters, a generalized Hankel matrix can be constructed, and the complete bilinear realization can be achieved as in the following section.

4.6.2. The bilinear Hankel matrix

In the linear case, the Hankel matrix is defined as the product between the observability and the reachability matrices [92]. Similarly, for the bilinear case, the bilinear Hankel matrix denoted as $\mathcal{H}$ is defined accordingly and is the following double infinite operator:

$$\mathcal{H} = \mathcal{Q}\mathcal{P} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \vdots \end{bmatrix} \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \cdots \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1\mathbf{P}_1 & \mathbf{Q}_1\mathbf{P}_2 & \cdots \\ \mathbf{Q}_2\mathbf{P}_1 & \mathbf{Q}_2\mathbf{P}_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{4.109}$$

The above equation is in a block Hankel structure representation, so by expanding the computations to each block, we can rewrite it as:

$$\mathcal{H} = \mathcal{Q}\mathcal{P} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CN} \\ \vdots \end{bmatrix} \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{NB} & \cdots \end{bmatrix} = \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \mathbf{CNB} & \cdots \\ \mathbf{CAB} & \mathbf{CA}^2\mathbf{B} & \mathbf{CANB} & \cdots \\ \mathbf{CNB} & \mathbf{CNAB} & \mathbf{CN}^2\mathbf{B} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{4.110}$$

The bilinear Hankel matrix in Eq. (4.110) reveals the connection with the bilinear Markov parameters $\mathcal{W} = \mathbf{C}\mathcal{P}$. In particular, the first row of $\mathcal{H}$ consists of the Markov parameters, and the first column is a mutation of the first row. As a result, the bilinear Hankel is not symmetric. Further, the inner blocks are constructed by appropriately reshuffling the first column or first row whenever the partnership exists under or upper the main diagonal. To illustrate the relation between the Hankel matrix and the Markov parameters for the bilinear case, we construct the following finite Hankel matrix $\mathbf{H} \in \mathbb{R}^{2^k-1\times 2^k-1}$ after knowing a finite set of the bilinear Markov parameters $\mathbf{W} \in \mathbb{R}^{1\times 2^k-1}$. For instance, if $k = 4$, then 15-Markov parameters are known and can fill out partial blocks of the potential Hankel matrix of size 15×15 with the fully completed Hankel to be of length 3×3 (roughly the floor of the square root of 15, which is 3).

$$\left[\begin{array}{l|c|cc|cccc|ccccc}
\mathbf{H = QP} & \mathbf{B} & \mathbf{AB} & \mathbf{NB} & \mathbf{A^2B} & \mathbf{ANB} & \mathbf{NAB} & \mathbf{N^2B} & \mathbf{A^3B} & \mathbf{A^2NB} & \mathbf{ANAB} & \mathbf{AN^2B} & \cdots \\
\hline
\mathbf{C} & * & * & * & * & * & * & * & * & * & * & * & \cdots \\
\hline
\mathbf{CA} & * & * & * & * & * & * & * & 0 & 0 & 0 & 0 & \cdots \\
\mathbf{CN} & * & * & * & * & * & * & * & 0 & 0 & 0 & 0 & \cdots \\
\hline
\mathbf{CA^2} & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mathbf{CNA} & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mathbf{CAN} & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\mathbf{CN^2} & * & * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\vdots & & & & & & & & & & & &
\end{array}\right] \tag{4.111}$$

The finite Hankel in Eq (4.111) results after filling it out with the bilinear Markov parameters. The first row and the first column are the *reachability-observability* matrices, respectively, where the Hankel is constructed after taking the product between them. The second row consists of the Markov parameters, and as we have assumed that 15 parameters are known, the star notation is used whenever the parameter is known. On the other hand, the zeros are not activated by Markov parameters as the simulation has stopped at $k = 4$. Therefore, the zero-block pattern allows one square to complete the Hankel matrix of dimension 3 in the upper left corner. In general, the construction of the bilinear Hankel matrix follows the next rule described in [95].

The first row ($i = 1$) consists of the Markov parameters.

$$\mathcal{S}_{1j} = \mathbf{w}_j,\ \ j = 1, 2, \ldots \tag{4.112}$$

For $i \geq 2$ and $j = 1, 2, \ldots$; the rest of the Hankel is constructed as:

$$\mathcal{S}_{i-1,j+1} = \left[\begin{array}{cc} \mathcal{S}^{(\mathbf{A})}_{i-1,j+1} & \mathcal{S}^{(\mathbf{N})}_{i-1,j+1} \end{array}\right] \tag{4.113}$$

Where the partition assigns the same number of columns to both blocks on the right-hand side and

$$\mathcal{S}_{ij} = \left[\begin{array}{c} \mathcal{S}^{(\mathbf{A})}_{i-1,j+1} \\ \mathcal{S}^{(\mathbf{N})}_{i-1,j+1} \end{array}\right]. \tag{4.114}$$

Finally, the Hankel matrix is

$$\mathcal{H} = \left[\begin{array}{ccc} \mathcal{S}_{11} & \mathcal{S}_{12} & \cdots \\ \mathcal{S}_{21} & \mathcal{S}_{22} & \cdots \\ \vdots & \vdots & \ddots \end{array}\right] \tag{4.115}$$

and is formed with the same elements as the definition provides.

Both previous sections have been introduced to connect the input-output data with constructing a generalized Hankel matrix that will lead to bilinear realization. Therefore, the following section provides all the relevant results presented in [95] and algorithms.

4.6.3. Bilinear realization algorithm

We state the two theorems from [95] that prove the partial and complete bilinear realization from the Markov parameter sequence.

Theorem 4.28 (Isidori [95]):
Let $\{w_1, w_2, \ldots, w_{M_0}\}$ be an arbitrary finite sequence of 2^{j-1} row vectors with $j = 1, \ldots, M_0$. Suppose there exist positive integers M and $M^{'} = M_0 - M$ such that:

$$\texttt{rank}\left(\mathcal{S}_{M'M}\right) = \texttt{rank}\left(\mathcal{S}_{M',M+1}\right) = \texttt{rank}\left(\mathcal{S}_{M'+1,M}\right)$$

is satisfied. In that case, the minimal partial bilinear realization problem has a unique solution (modulo the choice of bases in the state space). The dimension of the realization is equal to $\texttt{rank}(\mathcal{S}_{M'M})$, and an actual realization is provided by the following quadruplet Eqs. (4.116-4.119). ◊

Theorem 4.29 (Isidori [95]):
An arbitrary infinite sequence $\{w_1, w_2, \ldots, w_j, \ldots\}$ of 2^{j-1} row vectors admits a complete bilinear realization if the infinite matrix $\mathcal{H}$ has finite rank n. The dimension of the minimal realization is equal to n an actual minimal realization is provided by the quadruplet Eqs. (4.116-4.119). ◊

Let the singular value decomposition of the bilinear Hankel matrix be $\mathcal{H}_m = \mathbf{U\Sigma V}^T$. The quadruplet of dimension n is constructed as:

$$\begin{aligned}
\mathbf{A}_n &= \mathbf{\Sigma}^{-1/2}\mathbf{U}^T\mathbf{S}^{\mathbf{A}}\mathbf{V}\mathbf{\Sigma}^{-1/2} && (4.116)\\
\mathbf{N}_n &= \mathbf{\Sigma}^{-1/2}\mathbf{U}^T\mathbf{S}^{\mathbf{N}}\mathbf{V}\mathbf{\Sigma}^{-1/2} && (4.117)\\
\mathbf{B}_n &= \mathbf{\Sigma}^{1/2}\mathbf{V}^T \quad \rightarrow \quad \text{1st column} && (4.118)\\
\mathbf{C}_n &= \mathbf{U}\mathbf{\Sigma}^{1/2} \quad \rightarrow \quad \text{1st row} && (4.119)
\end{aligned}$$

The theoretical part for achieving bilinear realization by having the freedom of repeated simulations is completed. Therefore, we employ a toy bilinear system to illustrate the algorithm's applicability. In the latter, the novelty will be to relax some conditions. For instance, the following aspects will be resolved:

4.6.3.1. An introductory example

Example 4.6 (A toy system):
Let the following bilinear system of order 2 with matrices

$$\mathbf{A} = \begin{bmatrix} 0.9 & 0.0 \\ 0.0 & 0.8 \end{bmatrix},\ \mathbf{N} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix},\ \mathbf{C} = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}^T. \qquad (4.120)$$

Algorithm 4.2: Bilinear realization from input-output time domain data

```
Input: Input-output time-domain data from a system $u \rightarrow \boxed{\mathbf{\Sigma}?} \rightarrow y$.
Output: A minimal bilinear system $(\mathbf{A}_r, \mathbf{N}_r, \mathbf{B}_r, \mathbf{C}_r)$ of dimension $r$ s.t., $\mathbf{\Sigma}_r \approx \mathbf{\Sigma}$.
1: Excite the system $\mathbf{\Sigma}$ $k$-times with $\mathbf{u}_m \sim \mathcal{N}(\mu, \sigma)$ and collect $\mathbf{y}_m$, where $k = 2^{m-1}$.
     1st   $[u_1(1) \cdots u_1(m)] \rightarrow \boxed{\mathbf{\Sigma}} \rightarrow [y_1(1) \cdots y_1(m)] = \mathbf{Y}_1$, and $\mathbf{U}_1$(Definition 4.27).
      ⋮                                                  ⋮
     kth   $[u_k(1) \cdots u_k(m)] \rightarrow \boxed{\mathbf{\Sigma}} \rightarrow [y_k(1) \cdots y_k(m)] = \mathbf{Y}_k$, and $\mathbf{U}_k$ (Definition 4.27).
2: Identify the $(2^m - 1)$ bilinear Markov parameters by solving the linear system
   Eq. (4.107).
3: Construct the bilinear Hankel matrix $\mathcal{H}$ along with the partitions $\mathcal{S}^{\mathbf{A}}$, $\mathcal{S}^{\mathbf{N}}$ of
   dimension $p = 2^{m/2} - 1$.
4: Project with SVD the matrices $(\mathcal{H}, \mathcal{S}^{\mathbf{A}}, \mathcal{S}^{\mathbf{N}})$ as in equations (4.116-4.119) and
   construct the reduced/identified bilinear model $(\mathbf{A}_r, \mathbf{N}_r, \mathbf{B}_r, \mathbf{C}_r)$.
```

We start by estimating the bilinear Markov parameters. By choosing $m = 4$, we can recover $2^m - 1 = 15$ Markov parameters. The solution of the system in Eq. (4.107) gave:

$$\mathbf{W} = [\ 1.0 \quad 0.9 \quad 0.4 \quad 0.81 \quad 0.33 \quad 0.36 \quad 0.22 \quad 0.729 \quad 0.273 \quad 0.297 \quad 0.183 \quad 0.324 \quad 0.18 \quad 0.198 \quad 0.118\ ]$$

By reshuffling the vector $\mathbf{W}$ with the above rules, we can form the Hankel-$\mathbf{H}$ matrix of dimension $p = 2^{m/2} - 1 = 3$ along with the shifted versions $\mathbf{S}^{\mathbf{A}}$, $\mathbf{S}^{\mathbf{N}}$.

The completed Hankel matrix of dimension $p = 3$, along with the shifted version, are:

$$\mathbf{H} = \begin{bmatrix} 1.0 & 0.9 & 0.4 \\ 0.9 & 0.81 & 0.33 \\ 0.4 & 0.36 & 0.22 \end{bmatrix}, \ \mathbf{S}^{\mathbf{A}} = \begin{bmatrix} 0.9 & 0.81 & 0.33 \\ 0.81 & 0.729 & 0.273 \\ 0.36 & 0.324 & 0.18 \end{bmatrix}, \ \mathbf{S}^{\mathbf{N}} = \begin{bmatrix} 0.4 & 0.36 & 0.22 \\ 0.33 & 0.297 & 0.183 \\ 0.22 & 0.198 & 0.118 \end{bmatrix}.$$

The singular value decomposition has two significant aspects in this framework. Firstly, it provides a minimal degree of the bilinear model. Secondly, it constructs the identified or reduced model using the left and right singular vector matrices as projection matrices.

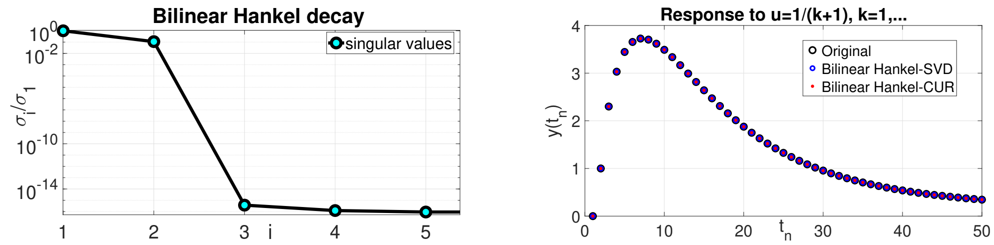


Figure 4.17.: Left: The singular values decay of the bilinear Hankel matrix is depicted. Right: With input response, $u_k = 1/(k+1)$, $k = 0, 1, \ldots$, all the models are equivalent.

In Fig. 4.17, the 3rd normalized singular value has reached machine precision $\sigma_3/\sigma_1 = 5.2501e - 17$, which is precisely the criterion for the minimality of the underline bilinear system. Therefore, we construct the bilinear model of order $r = 2$ and the results:

$$\mathbf{A}_r = \begin{bmatrix} 0.89394 & 0.11305 \\ 0.0050328 & 0.80606 \end{bmatrix}, \ \mathbf{N}_r = \begin{bmatrix} 0.41116 & -0.2281 \\ -0.24782 & 0.088841 \end{bmatrix}, \ \mathbf{B}_r = \begin{bmatrix} -1.0001 \\ -0.053577 \end{bmatrix}, \ \mathbf{C}_r = \begin{bmatrix} -1.0001 \\ 0.0040101 \end{bmatrix}^T. \quad (4.121)$$

The above system is an equivalent modulo with the original one, meaning there is a transformation $\mathbf{T}$ Appendix C that makes both systems to be aligned. Another way that the dynamical equivalence of these systems can be proved is by computing the eigenvalues of the linear and bilinear operators. Thus, easily can be checked that holds $\texttt{eig}(\mathbf{A}) = \texttt{eig}(\mathbf{A}_r)$ and $\texttt{eig}(\mathbf{N}) = \texttt{eig}(\mathbf{N}_r)$. ◊

4.6.3.2. A large scale example

Example 4.7 (Large scale example: The Burgers' equation):
In this section, we are concerned with the PDEs where the spatial discretized version of the continuous models can easily reach the large-scale setting. By varying specific parameters, e.g., viscosity, we conclude the applicability of the proposed reduction method near the point where the parabolic structure of a PDE can transform into a hyperbolic problem where conventional MOR techniques start to break down.

A well-studied nonlinear example is that of the Burgers' Eq. (4.122). The Burgers' PDE occurs in various areas of applied mathematics, such as fluid mechanics, nonlinear acoustics, etc. For a given field $u(x,t)$ and diffusion coefficient (or kinematic viscosity) ν, the general form of the viscous Burgers' equation in one space dimension is the following dissipative system:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}. \tag{4.122}$$

When the diffusion term is absent (i.e., $\nu = 0$), Burgers' equation becomes the inviscid Burgers' equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0, \tag{4.123}$$

Which is a prototype for conservation equations that can develop discontinuities (shock waves). In particular, Eq. (4.123) is the advective form of the Burgers' equation.

Following Breiten and Damm, after spatial semi-discretization of this nonlinear partial differential equation using k nodes in a finite difference scheme, we end up with an ordinary differential equation including a quadratic nonlinearity. Further, we can approximate this system using the Carleman linearization technique. Hence, we use a second-order approximation that yields a linearized system of dimension $n = k + k^2$.

The fully discretized model can be achieved by enforcing time discretization. Introducing the backward Euler scheme with time step dt, the derivative approximation is $\dot{\mathbf{x}}(t) \approx (\mathbf{x}(t) - \mathbf{x}(t-dt))dt^{-1}$, where the same linear relations can be found in [23]:

Remark 4.30 (Backward Euler time discretization for bilinear systems):
The continuous in-time Burgers' model can be discretized in time with the backward Euler as:

$$\mathbf{A}_d = (\mathbf{I} - dt\mathbf{A}_c)^{-1}, \ \mathbf{N}_d = dt(\mathbf{I} - dt\mathbf{A}_c)^{-1}\mathbf{N}_c, \ \ \mathbf{B}_d = dt(\mathbf{I} - dt\mathbf{A}_c)^{-1}\mathbf{B}_c, \ \ \mathbf{C}_d = \mathbf{C}_c. \tag{4.124}$$

It is also important that the above discretization scheme is bijective; therefore, by having the discrete in-time operators, we can infer the continuous operators as well from:

$$\mathbf{A}_c = \frac{1}{dt}(\mathbf{I} - \mathbf{A}_d^{-1}),\ \mathbf{N}_c = \left(\frac{1}{dt}\mathbf{I} - \mathbf{A}_c\right)\mathbf{N}_d,\ \mathbf{B}_c = \left(\frac{1}{dt}\mathbf{I} - \mathbf{A}_c\right)\mathbf{B}_d,\ \mathbf{C}_c = \mathbf{C}_d. \tag{4.125}$$

◊

Specifying a sampling step dt, we want to reconstruct the continuous-time response from a sample sequence, assuming one sample per time interval with width dt. Similarly, with a zero-order hold (ZOH). Moreover, the Nyquist-Shannon sampling theorem is a fundamental bridge between continuous-time and discrete-time signals. In particular, the theorem establishes a sufficient condition for a sample rate that permits a discrete sequence of samples to capture all the information from a continuous-time signal of finite bandwidth. For the Burgers' example, we investigate the behavior by varying the discretization parameters and the viscosity up to the point where the hyperbolic profile of the PDE (e.g., $\nu \to 0$) produces shock waves.

For the FOM simulation, we use $\nu = 0.1$ and $k = 30$, so we have to deal with a continuous bilinear system of type as in Eq. (4.101), and dimension 930 where after the above discretization in Eq. (4.124), the discretized bilinear system is of the same dimension and as in Eq. (4.102). The aim is to construct from a finite set of input-output time-domain measurements a continuous-time nonlinear model of bilinear type with low dimension capable of accurately approximating the FOM.

Simulating with a white input of length $m = 10$, e.g., $\mathbf{u} = \begin{bmatrix} u_1 & \cdots & u_{10} \end{bmatrix}$ with $dt = 0.1$, gives an output of the same length, e.g., $\mathbf{y} = \begin{bmatrix} y_1 & \cdots & y_{10} \end{bmatrix}$. Repeating the simulation for $k = 2^{m-1} = 512$ times, the least square problem obtains the unique solution consisting of $2^m - 1 = 1023$ bilinear Markov parameters. Therefore, as described before, constructing a squared bilinear Hankel matrix of size $p = 31$ is possible.

Continuously, the decay of the singular values of the bilinear Hankel matrix in Fig. 4.18(left) allows the specification of the order that the reduced model will have in terms of trading the complexity with the accuracy. The linear Hankel decay is also depicted in Fig. 4.18(left). The linear decay can be obtained after simulating the bilinear system with zero initial conditions and applying an impulse input. In such a case, the bilinear term will not be activated; therefore, the output will give the impulse response of the linear subsystem. Also, this indicates that someone to check the bilinear nonlinearity would need persistent excitation inputs. The recommended reduced bilinear model is of

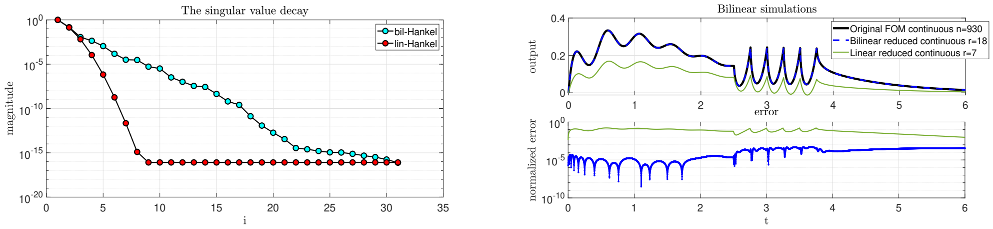


Figure 4.18.: The singular value decay of the bilinear and linear Hankel frameworks. Right: The Burgers' equation under the same input forcing in Fig. 4.7.

order $r = 18$ where the 19th normalized singular holds $\sigma_{19}/\sigma_1 = 1.1820e-12$. Evaluating the discrete bilinear ROM of order $r = 18$, we can compare it with the original discrete FOM under the following excitation $u_k = 1/(k+1),\ k = 0, 1, \ldots$.

In Fig. 4.18(right), the reduced continuous model of order $r = 18$ obtained from input-to-output time-domain data lies on top of the original FOM with dimension 930 with an error performance around the $O(1e-5)$. The inverse transformations from the discrete to the continuous operators in Eq. (4.125) are performed to derive the continuous bilinear model. Finally, the bilinear fit is compared with the linear fit obtained from the linear Hankel framework, where it is evident that it cannot capture the nonlinear behavior of the Burgers' equation. ◊

4.7. From a single i/o data sequence to bilinear realization

In many cases, data from a simulated system are available as a unique sequence [119]. To achieve bilinear realization as in [95], many repetitive data assimilation simulations in the time domain should be performed. Therefore, in the case of a single experiment, the idea is to learn a NARX model from this unique data sequence capable of predicting the output behavior under different excitations. The NARX model will play the role of the original simulator, which might be unavailable for several simulations or the process is very expensive. Then, by constructing such a model and combining the realization theory in [95], a state-space bilinear model can be constructed as in Eq. (4.101). Using a state-space model is beneficial compared to the NARX as it relies on the classical nonlinear control theory with many results, especially on bilinear systems in the direction of stability and control.

In real-life applications, the data set is corrupted with noise. Therefore, regression techniques are suitable as the noise can be filtered efficiently. Machine learning regression techniques rely on this aspect as well. Next, we will demonstrate two ways of building a model that will serve the scope of the unknown simulator. The first one is based on the neural architecture where the coefficients/parameters of the model have nonlinear dependency due to the multi-composition scheme that describes a Neural Network (NN). The NN design can approximate strong nonlinearities when the data availability is large. Still, on the other hand, the generalization of the model and the over-fitting control rely heavily on regularization techniques where handling those features becomes easily ad-hoc and depends strongly on the application. Another type of nonlinear autoregressive model is the moving average NARMAX model, which has similar regression behavior with the NARX net, but the NARMAX is linear to the parameters. In this case, the NARMAX is simpler than the NARX net, and the capability of capturing nonlinear features for many cases is questionable.

4.7.1. Nonlinear autoregressive models with exogenous inputs

A general discrete bilinear input-output model, as stated in [45], takes the NARMAX (Non-linear AutoRegressive Moving Average with eXogenous inputs) form

$$y(k) = q + du(k) + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + \sum_{i=1}^{n_y}\sum_{j=1}^{n_u} c_{ij} y(k-i)u(k-j). \quad (4.126)$$

How well does the above model approximate nonlinear systems, particularly the bilinear? For linear discrete-time systems, it is well known that a linear difference equation model exists that involves only a fixed and finite number of calculations at each stage if the Hankel matrix of the system has a finite rank, which provides a much more concise description than the impulse response function. An analogous situation exists for non-linear discrete-time systems. The NARMAX (Leontaridis and Billings 1985) provides a unified representation for a broad class of non-linear systems and has apparent advantages over functional series representation such as the Volterra series. Based on the Stone-Weierstrass theorem, two models, the output-affine and polynomial models, are suitable for this purpose. These results should provide a basis for using these two models to identify non-linear systems and other control implementations.

The model

$$y(k) = F\left(y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)\right) \quad (4.127)$$

where $F(\cdot)$ is some non-linear function, is about as far as one can go in specifying a general finite non-linear system. Leontaridis and Billings (1985) proved that a nonlinear discrete-time time-invariant system can always be represented by the model in a region around an equilibrium point subject to two sufficient conditions:

1. the response function f of the system is finitely realizable;
2. a linearized model exists if the system is operated close to the chosen equilibrium point.

4.7.2. A real engineering example

Example 4.8 (Heat exchanger):
The process is a liquid-saturated steam heat exchanger, where water is heated by pressurized saturated steam through a copper tube. The input variable is the liquid flow rate, and the output variable is the outlet liquid temperature. The sampling time is 1s, and the number of samples is $4,000$. Data can be downloaded from the database to identify systems (DaISy): `https://homes.esat.kuleuven.be/~tokka/daisydata.html`. The following data set is depicted in Fig. 4.19.

After detrending the data, the next step is to fit a NARMAX model of Eq. (4.126) with maximum delay (lag=5). We can use the whole data set, but here, we choose the same subset of $1,000$ points as in [119], which also serves the validation scope.

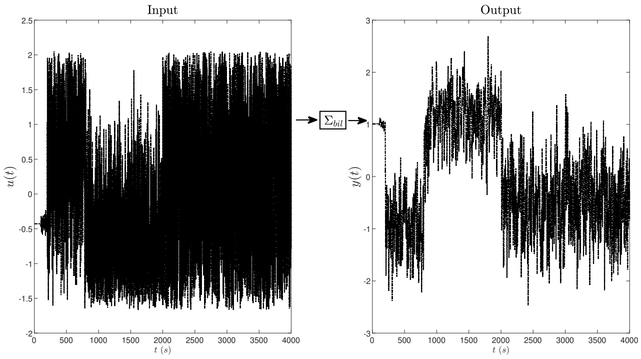


Figure 4.19.: Input-output data from DaISy.

The NARMAX structure fitted above can explain the data under some errors. However, the idea is to use the NARX-net as a more general class than that of NARMAX models, which can eventually cover the bilinear dynamics if enough neurons and layers have been assumed. Figure 4.20) illustrates the superiority of the proposed method in terms of performance. From the single i/o data sequence, a NN with 3-layers and 20-lags was trained using the same training data[4] as in [119] (1000 points). In addition, the trained NN was used in the bilinear realization algorithm to generate more data. A stable reduced bilinear model of order $r = 3$ shown in (4.128) was successfully constructed. The original noisy data were explained with a lower mean percentage error MPE $= 0.56\%$ compared to the subspace method for the entire dataset. Another NN architecture, s.a., the NARMAX[5] belongs to a subclass of bilinear systems, and will filter some nonlinear features without achieving such a good MPE.

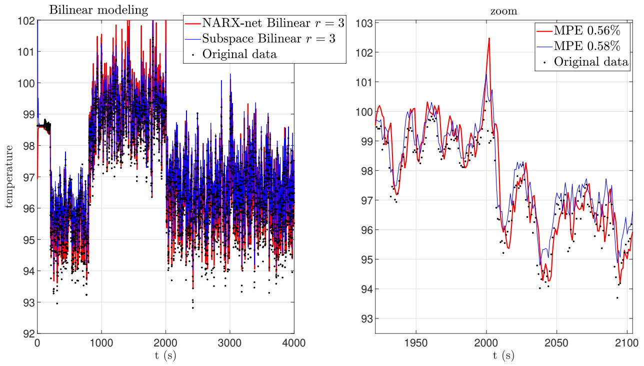


Figure 4.20.: Comparison and model fit of the proposed NARX-net bilinear model (4.128) with the subspace method from [119] for the same reduced order ($r = 3$).

[4] **Data detrend**: $u_n = (u - \bar{u})/\sigma_u,\ y_n = (y - \bar{y})/\sigma_y$; **zero-response**: data were doubled in size for learning the zero-response, i.e., $u_n = 0 \rightarrow \boxed{\boldsymbol{\Sigma}} \rightarrow y_n = 0$.

[5] **NARMAX**: The nonlinear autoregressive moving average model with eXogenous input [45, 3].

$$\begin{cases} \dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} 0.9164 & 0.09167 & -0.1847 \\ -0.2663 & -0.1515 & 0.1232 \\ -0.07227 & 0.4778 & 0.3571 \end{bmatrix}}_{\mathbf{A}_{id}} \mathbf{x}(t) + \\ \quad \underbrace{\begin{bmatrix} 0.02717 & 0.5169 & 0.5555 \\ -0.09674 & 0.5467 & 0.5696 \\ 0.1878 & -0.06846 & -1.981 \end{bmatrix}}_{\mathbf{N}_{id}} \mathbf{x}(t)u(t) + \underbrace{\begin{bmatrix} 2.9063 \\ 2.909 \\ -0.16088 \end{bmatrix}}_{\mathbf{B}_{id}} u(t) + \underbrace{\begin{bmatrix} -1.073 \\ -1.074 \\ 0.05938 \end{bmatrix}}_{\mathbf{L}_{id}}, \\ y(t) = \underbrace{\begin{bmatrix} -0.7852 & 0.7794 & -0.05203 \end{bmatrix}}_{\mathbf{C}_{id}} \mathbf{x}(t) + \underbrace{96.9358}_{\mathbf{T}_{id}}, \ \mathbf{x}(0) = \mathbf{0}, \ t \geq 0. \end{cases} \tag{4.128}$$

$\Diamond$

In conclusion, NN architectures are a superclass of NARMAX models used in the classical identification theory. Consequently, NN models share the same strong argument with the Carleman linearization scheme that can approximate general nonlinear systems. Finally, NN and realization theory successfully bridge data science with computational science to build reliable, interpretable nonlinear models. Different NN architectures (s.a., recurrent NNs) in combination with other realization frameworks (s.a., the Loewner framework) and for other types of nonlinearities

CHAPTER 5

IDENTIFICATION AND REDUCTION OF QUADRATIC-BILINEAR SYSTEMS

Contents

5.1. Introduction

This chapter presents non-intrusive, i.e., purely data-driven methods using the Loewner framework (LF) and nonlinear optimization techniques to identify or reduce quadratic-bilinear control systems from input-output (i/o) time-domain measurements. At the heart of these methods are optimization schemes that enforce interpolation of the symmetric generalized frequency response functions (GFRFs) as derived in the Volterra series framework. We consider harmonic input excitations to infer such measurements. After reaching the steady-state profile, the symmetric GFRFs can be measured from the Fourier spectrum (phase and amplitude). Using these measurements properly for the pure quadratic case (zero bilinear terms), we can identify low-order nonlinear state-space models with non-trivial equilibrium state points in the quadratic form, such as the Lorenz attractor. In particular, for the multi-point equilibrium case, where measurements describe some local bifurcated models to different coordinates, we achieve global model identification after solving an operator alignment problem based on a constrained quadratic matrix equation. We test the new method for a more demanding system in state dimension, i.e., the viscous Burgers' equation with Robin boundary conditions. The complexity reduction and approximation accuracy are tested.

In the first part of this chapter, we are ready to introduce and analyze our new method that uses i/o time-domain data to identify or construct quadratic state-space models after combining the Loewner and Volterra frameworks with nonlinear optimization techniques. Compared to our prior work and previous chapters in this thesis [104, 105, 72], the advances are; 1) we use generalized frequency measurements from higher orders kernels of the symmetric type that explain the propagating harmonics in the time domain output (measurable), making the reverse process (from time to frequency) feasible through the Fourier transform; 2) we solve the resulting nonlinear optimization problems to achieve quadratic model construction that interpolates the Volterra series to more kernels; 3) we identify global quadratic systems of low order with nontrivial equilibrium points after measuring the local dynamical behavior and solving a nonlinear matrix equation; 4) we test the proposed method for both scopes of identification and reduction with classical benchmarks such as the forced Lorenz attractor and the viscous Burgers' equation.

In the remaining chapter, we propose a similar data-driven procedure to fit quadratic-bilinear surrogate models from data in the quadratic bilinear part. Although the dynamics characterizing the original model are strongly nonlinear, we rely on lifting techniques to embed the original model into a quadratic-bilinear format. Here, data represent generalized transfer function values.This method is an extension of methods that do bilinear or quadratic inference separately.. It is based on fitting a linear model with the classical Loewner framework and then inferring the best supplementing nonlinear operators in a least-squares sense. Electrical circuits give the application scope of this method with nonlinear components (such as diodes). We propose various test cases to illustrate the performance of the method.

Finally, we provide a small section with the spectral identification method for the quadratic-bilinear system's case. Thus, identification from state access data that produces models independent of the input, along with an example, is presented.

5.2. Continuous-time quadratic control systems

We continue our analysis with the general state-space representation of a system in the quadratic form and for the SISO case:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{Q}(\mathbf{x}(t) \otimes \mathbf{x}(t)) + \mathbf{B}u(t), \\ y(t) = \mathbf{C}\mathbf{x}(t),\ \mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0},\ t \geq 0, \end{cases} \tag{5.1}$$

where the state-dimension is n, and the operators are: $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{Q} \in \mathbb{R}^{n\times n^2}$, $\mathbf{B}, \mathbf{C}^T \in \mathbb{R}^{n\times 1}$. The Kronecker product $\otimes$ is defined as in the following simple case $\begin{bmatrix} x_1 & x_2 \end{bmatrix} \otimes \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} x_1^2 & x_1x_2 & x_2x_1 & x_2^2 \end{bmatrix}$. Due to the commutative property, the matrix $\mathbf{Q}$ denotes the Hessian of the right-hand side and exhibits a particular symmetric structure. For two arbitrary vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, we can always ensure that it holds

$$\mathbf{Q}(\mathbf{u} \otimes \mathbf{v}) = \mathbf{Q}(\mathbf{v} \otimes \mathbf{u}). \tag{5.2}$$

Similarly, we want representations of the underlying nonlinear system in both time and frequency domains as in the linear case. As we have exploited tools such as the Volterra series expansion for approximating general nonlinear systems, we now focus on enforcing the structure of the quadratic state-space model. The first aim is to derive the symmetric GFRFs for the quadratic case that can be processed from the time domain to the frequency domain, and the second is to use these measurements to identify the hidden operators $(\mathbf{A},\ \mathbf{Q},\ \mathbf{B},\ \mathbf{C})$.

5.2.1. Higher-order transfer functions for the quadratic control system

The Volterra series Eq. (2.28) describes the approximation of nonlinear systems through higher-order generalized kernels in a multi-convolutional scheme. As explained in [137], different ways of extracting kernels exist. One way is the variational approach, where the structure of the triangular (or regular kernels) can be revealed through Picard iterations. In particular, the regular kernels can be derived after shifting the frequency domain of the triangular kernels. The regular kernels are convenient due to the asymmetric structure that makes them valuable for interpolation frameworks such as the Loewner and its nonlinear extensions. Despite the intrusive ease of use, the regular Volterra kernels cannot be measured directly from the time domain. Therefore, we choose another way of deriving higher-order Volterra kernels, namely the growing exponential approach (e.g., the probing method) for treating the issue of kernel estimation. With probing (harmonic excitation) of the system, and after processing the steady-state time evolution in the frequency domain via the Fourier (a special case of Laplace over the imaginary axis) transform, the time domain signal is decomposed to harmonics that scale and shift w.r.t. the symmetric GFRFs. Methods for estimating these symmetric kernels (e.g., kernel separation) were introduced in [37, 104, 150]. In the Appendix A, we analytically derive the three first symmetric generalized frequency response functions with the probing method. These are summarized here

$$
\begin{aligned}
H_1(s_1) &= \mathbf{C}\underbrace{(s_1\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}}_{\mathbf{G}_1(s_1)},\\
H_2(s_1,s_2) &= \frac{1}{2}\mathbf{C}\underbrace{\left((s_1+s_2)\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{Q}\left(\mathbf{G}_1(s_1)\otimes\mathbf{G}_1(s_2)+\mathbf{G}_1(s_2)\otimes\mathbf{G}_1(s_1)\right)}_{\mathbf{G}_2(s_1,s_2)}\\
H_3(s_1,s_2,s_3) &= \frac{1}{6}\mathbf{C}\left((s_1+s_2+s_3)\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{Q}\times\cdots\\
&\cdots\times\left[\mathbf{G}_1(s_1)\otimes\mathbf{G}_2(s_2,s_3)+\mathbf{G}_2(s_2,s_3)\otimes\mathbf{G}_1(s_1)+\right.\\
&\quad\mathbf{G}_1(s_2)\otimes\mathbf{G}_2(s_1,s_3)+\mathbf{G}_2(s_1,s_3)\otimes\mathbf{G}_1(s_2)+\\
&\quad\left.\mathbf{G}_1(s_3)\otimes\mathbf{G}_2(s_1,s_2)+\mathbf{G}_2(s_1,s_2)\otimes\mathbf{G}_1(s_3)\right].
\end{aligned}
\tag{5.3}
$$

At this point, we illustrate some of the properties the derived symmetric transfer functions inherit for the quadratic control system case.

- **Symmetry**: As it is evident, any permutation of the set $(s_1, s_2, \ldots, s_n)$ will result to the same evaluation of the $H_n(s_1, s_2, \ldots, s_n)$ and $\mathbf{G}_n(s_1, s_2, \ldots, s_n)$.

- **Decompositions**: Introducing the general reachability $\mathcal{R}$ and observability $\mathcal{O}$ counterparts can enable a more concise representation of the kernels. As we have introduced the reachability matrices $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_3$, here are the corresponding observability matrices:

$$
\begin{aligned}
\mathbf{O}_1(s_1) &= \mathbf{C}\boldsymbol{\Phi}(s_1),\\
\mathbf{O}_2(s_1,s_2) &= \frac{1}{2}\mathbf{C}\boldsymbol{\Phi}(s_1+s_2),\\
\mathbf{O}_3(s_1,s_2,s_3) &= \frac{1}{6}\mathbf{C}\boldsymbol{\Phi}(s_1+s_2+s_3).
\end{aligned}
\tag{5.4}
$$

Next, we introduce the following Table 5.1 that illustrates the dependencies of the quadratic operator as these are decomposed in observability and reachability counterparts. With the above observations and notations, we can derive a more convenient representation of the input to state GFRFs by exploiting their structure and stressing the positioning of the quadratic operator as in the 3rd level with the

input-output GFRF	$\mathcal{O}$	$\mathcal{R}$
$H_1(s_1)$	$\mathbf{O}_1(s_1)$	$\mathbf{R}_1 = \mathbf{B}$
$H_2(s_1, s_2, \mathbf{Q})$	$\mathbf{O}_2(s_1, s_2)$	$\mathbf{R}_2(s_1, s_2)$
$H_3(s_1, s_2, s_3, \mathbf{Q})$	$\mathbf{O}_3(s_1, s_2, s_3)$	$\mathbf{R}_3(s_1, s_2, s_3, \mathbf{Q})$

Table 5.1.: Quadratic operator dependency over the input to state kernels concerning the generalized controllability and observability counterparts.

superscripts $(\cdot)^\ell$-left and $(\cdot)^r$-right.

$$\begin{aligned}\mathbf{G}_1(s_1) &= \mathbf{\Phi}(s_1)\mathbf{R}_1,\\ \mathbf{G}_2(s_1,s_2,\mathbf{Q}) &= \frac{1}{2}\mathbf{\Phi}(s_1,s_2)\mathbf{Q}\mathbf{R}_2(s_1,s_2),\\ \mathbf{G}_3(s_1,s_2,s_3,\mathbf{Q}^\ell,\mathbf{Q}^r) &= \frac{1}{6}\mathbf{\Phi}(s_1,s_2,s_3)\mathbf{Q}^\ell\mathbf{R}_3(s_1,s_2,s_3,\mathbf{Q}^r)\end{aligned} \tag{5.5}$$

and for the input to output GFRFs as

$$\begin{aligned}H_1(s_1) &= \mathbf{O}_1(s_1)\mathbf{R}_1,\\ H_2(s_1,s_2,\mathbf{Q}) &= \mathbf{O}_2(s_1,s_2)\mathbf{Q}\mathbf{R}_2(s_1,s_2),\\ H_3^{\ell r}(s_1,s_2,s_3,\mathbf{Q}^\ell,\mathbf{Q}^r) &= \mathbf{O}_3(s_1,s_2,s_3)\mathbf{Q}^\ell\mathbf{R}_3(s_1,s_2,s_3,\mathbf{Q}^r).\end{aligned} \tag{5.6}$$

- **The reachability matrix $\mathbf{R}_3(\mathbf{Q})$ is linear w.r.t the quadratic operator $\mathbf{Q}$.** Assume $\lambda_1,\ \lambda_2 \in \mathbb{R}$ and $\mathbf{Q}_1,\ \mathbf{Q}_2 \in \mathbb{R}^{n\times n^2}$. Then, it holds
  - **Linear property: $\mathbf{R}_3(\lambda_1\mathbf{Q}_1 + \lambda_2\mathbf{Q}_2) = \lambda_1\mathbf{R}_3(\mathbf{Q}_1) + \lambda_2\mathbf{R}_3(\mathbf{Q}_2)$.**

    Proof. By neglecting the similar-structured terms (s.s.t), we can prove the following:

$$\begin{aligned}&\mathbf{R}_3(s_1,s_2,s_3,\lambda_1\mathbf{Q}_1+\lambda_2\mathbf{Q}_2) = \mathbf{G}_1(s_1)\otimes\mathbf{G}_2(s_2,s_3,\lambda_1\mathbf{Q}_1+\lambda_2\mathbf{Q}_2) + s.s.t.\\ &= \mathbf{G}_1(s_1)\otimes\frac{1}{2}\mathbf{\Phi}(s_1,s_2)(\lambda_1\mathbf{Q}_1+\lambda_2\mathbf{Q}_2)\mathbf{R}_2(s_1,s_2) + s.s.t.\\ &= \mathbf{G}_1(s_1)\otimes\frac{1}{2}\mathbf{\Phi}(s_1,s_2)\lambda_1\mathbf{Q}_1\mathbf{R}_2(s_1,s_2) + \mathbf{G}_1(s_1)\otimes\frac{1}{2}\mathbf{\Phi}(s_1,s_2)\lambda_2\mathbf{Q}_2\mathbf{R}_2(s_1,s_2) + s.s.t.\\ &= \lambda_1\mathbf{G}_1(s_1)\otimes\mathbf{G}_2(s_2,s_3,\mathbf{Q}_1) + \lambda_2\mathbf{G}_1(s_1)\otimes\mathbf{G}_2(s_2,s_3,\mathbf{Q}_2) + s.s.t.\\ &= \lambda_1\mathbf{R}_3(s_1,s_2,s_3,\mathbf{Q}_1) + \lambda_2\mathbf{R}_3(s_1,s_2,s_3,\mathbf{Q}_2).\end{aligned}$$

    □

Starting from the original dynamical system in Eq. (5.1) with the quadratic nonlinearity, we have derived all the quantities of interest with their properties for setting up our method. Equivalent descriptions between the time and frequency domain representations have been addressed for this problem using the Volterra theory.

5.3. Quadratic modeling from i/o time domain data

Next, we introduce the proposed method for computing quadratic state-space models from the first three symmetric GFRFs, which can be measured from time-domain harmonic excitation.

5.3.1. Identification of the linear subsystem with the Loewner framework

Using measurements of the 1st harmonic, we can identify the minimal linear subsystem of order $r \le n$, with an invertible $\hat{\mathbf{E}}$ as $(\hat{\mathbf{A}},\ \hat{\mathbf{B}},\ \hat{\mathbf{C}})$ with the Loewner framework. Further, by accessing the identified linear subsystem, we can formulate optimization problems where estimations of the quadratic operator can be achieved after using higher harmonics (kernels) information. We acquire and solve these optimization problems in two steps: solving an under-determined linear optimization problem in a least-squares setting and solving a non-linear optimization problem with the Newton method.

5.3.2. Estimation of the quadratic operator from the 2nd kernel

Identification of the minimal linear subsystem $(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}})$ of order r as described in Section 5.3.1 allows the construction of the reduced resolvent $\hat{\mathbf{\Phi}}(s) = (s\hat{\mathbf{I}} - \hat{\mathbf{A}})^{-1} \in \mathbb{C}^{r\times r}$, and the 2nd GFRFs with the unknown operator $\hat{\mathbf{Q}}$ can be written as:

$$\begin{aligned} \hat{H}_2(s_1,s_2) &= \underbrace{\frac{1}{2}\hat{\mathbf{C}}\hat{\mathbf{\Phi}}(s_1+s_2)}_{\hat{\mathbf{O}}_2(s_1,s_2)}\,\hat{\mathbf{Q}}\,\underbrace{\left[\hat{\mathbf{\Phi}}(s_1)\hat{\mathbf{B}}\otimes\hat{\mathbf{\Phi}}(s_2)\hat{\mathbf{B}}+\hat{\mathbf{\Phi}}(s_2)\hat{\mathbf{B}}\otimes\hat{\mathbf{\Phi}}(s_1)\hat{\mathbf{B}}\right]}_{\hat{\mathbf{R}}_2(s_1,s_2)} = \\ &= \hat{\mathbf{O}}_2(s_1,s_2)\hat{\mathbf{Q}}\hat{\mathbf{R}}_2(s_1,s_2) = \left(\hat{\mathbf{O}}_2(s_1,s_2)\otimes\hat{\mathbf{R}}_2^T(s_1,s_2)\right)\mathtt{vec}(\hat{\mathbf{Q}}). \end{aligned} \tag{5.7}$$

The way of estimating the quadratic operator $\hat{\mathbf{Q}}$ comes after enforcing interpolation with the 2nd harmonic (2nd kernel) over a 2D grid of selected measurements $\left(s_1^{(k)}, s_2^{(k)}\right)$. Thus, we enforce

$$\underbrace{H_2\left(s_1^{(k)}, s_2^{(k)}\right)}_{\text{k: measurements}} = \hat{H}_2\left(s_1^{(k)}, s_2^{(k)}\right), \tag{5.8}$$

and we construct the following solvable linear optimization problem by minimizing the 2-norm (least-squares) similarly to the quadratic-bilinear case in [105]. Collecting k pairs of measurements $(s_1^{(k)}, s_2^{(k)})$, we conclude that:

$$\underbrace{\begin{bmatrix} H_2(s_1^{(1)}, s_2^{(1)}) \\ H_2(s_1^{(2)}, s_2^{(2)}) \\ \vdots \\ H_2(s_1^{(k)}, s_2^{(k)}) \end{bmatrix}}_{\mathbf{Y}:\ (k\times 1)} = \underbrace{\begin{bmatrix} \hat{\mathbf{O}}_2^{(1)}\otimes\hat{\mathbf{R}}_2^{T(1)} \\ \hat{\mathbf{O}}_2^{(2)}\otimes\hat{\mathbf{R}}_2^{T(2)} \\ \vdots \\ \hat{\mathbf{O}}_2^{(k)}\otimes\hat{\mathbf{R}}_2^{T(k)} \end{bmatrix}}_{\mathbf{M}:\ (k\times r^3)} \underbrace{\mathtt{vec}(\hat{\mathbf{Q}})}_{r^3\times 1} \tag{5.9}$$

The quadratic operator inherits symmetries, e.g., the terms x_ix_j and x_jx_i appear twice in the product $\mathbf{x}\otimes\mathbf{x}$. These symmetries are known by construction Eq. (5.2) and can be handled properly. Nevertheless, taking care of these symmetries, the quadratic operator is not a unique representation of the original system. Its entries are not fully detectable using only information from the 2nd kernel. Algebraically, this can be explained by the rank deficiency of the least squares matrix $\mathbf{M} \in \mathbb{R}^{k\times r^3}$. Further, real symmetry can be enforced in Eq. (5.9) by including the conjugate counterparts. The above problem motivates the usage of higher harmonics (kernels) where the remaining parameters of the above under-determined problem can be estimated. In particular, evaluating the quadratic operator $\hat{\mathbf{Q}}$, can be parameterized further with the non-empty null space we have computed from the above least-squares problem.

The quadratic operator has r^3 unknowns (less due to symmetries). If the rank of the matrix $\mathbf{M}$ is $\mathtt{rank}(\mathbf{M}) = p < r^3$, the parametric solution of $\hat{\mathbf{Q}}$ that we obtain from H_2 measurements with the dimension of the kernel $m = r^3 - p$ can be written as:

$$\hat{\mathbf{Q}} = \hat{\mathbf{Q}}_s + \hat{\mathbf{Q}}_k = \underbrace{\hat{\mathbf{Q}}_s}_{\text{rank solution}} + \underbrace{\sum_{i=1}^{m} \lambda_i \hat{\mathbf{Q}}_i}_{\text{parameterization}} \tag{5.10}$$

The above splitting Eq. (5.10) can be considered the same when the operators $\hat{\mathbf{Q}}_s$, $\hat{\mathbf{Q}}_i$, $i = 1, \ldots, m$ are represented as vectors after vectorization due to the linear property of $\mathtt{vec}(\cdot)$[1].

5.3.3. Identification of the quadratic operator from the 3rd kernel

From the parameters λ_i in Eq. (5.10), we also search those that explain the 3rd kernel's interpolation. Therefore, we can write:

$$\hat{H}_3(s_1, s_2, s_3) = \hat{\mathbf{O}}_3(s_1, s_2, s_3)\hat{\mathbf{Q}}\hat{\mathbf{R}}_3(s_1, s_2, s_3, \hat{\mathbf{Q}}), \tag{5.11}$$

and substituting Eq. (5.10) in Eq. (5.11), due to the linear property of the operator $\mathbf{R}_3$ as explained in Section 5.2.1, we can derive

$$\begin{aligned}
&\hat{H}_3(s_1, s_2, s_3) = \hat{\mathbf{O}}_3(s_1, s_2, s_3)\left(\hat{\mathbf{Q}}_s + \sum_{i=1}^{m} \lambda_i \hat{\mathbf{Q}}_i\right) \hat{\mathbf{R}}_3 \left(s_1, s_2, s_3, \hat{\mathbf{Q}}_s + \sum_{i=1}^{m} \lambda_i \hat{\mathbf{Q}}_i\right) = \\
&= \hat{\mathbf{O}}_3(s_1, s_2, s_3)\hat{\mathbf{Q}}_s\hat{\mathbf{R}}_3\left(s_1, s_2, s_3, \hat{\mathbf{Q}}_s\right) + \hat{\mathbf{O}}_3(s_1, s_2, s_3)\hat{\mathbf{Q}}_s\hat{\mathbf{R}}_3\left(s_1, s_2, s_3, \sum_{i=1}^{m} \lambda_i \hat{\mathbf{Q}}_i\right) + \\
&\hat{\mathbf{O}}_3(s_1, s_2, s_3)\left(\sum_{i=1}^{m} \lambda_i \hat{\mathbf{Q}}_i\right) \hat{\mathbf{R}}_3\left(s_1, s_2, s_3, \hat{\mathbf{Q}}_s\right) + \hat{\mathbf{O}}_3(s_1, s_2, s_3)\left(\sum_{i=1}^{m} \lambda_i \hat{\mathbf{Q}}_i\right) \hat{\mathbf{R}}_3\left(s_1, s_2, s_3, \sum_{i=1}^{m} \lambda_i \hat{\mathbf{Q}}_i\right) \\
&= \hat{H}_3^{ss}(s_1, s_2, s_3) + \sum_{i=1}^{m} \lambda_i \left(\hat{H}_3^{is}(s_1, s_2, s_3) + \hat{H}_3^{si}(s_1, s_2, s_3)\right) + \sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_i \lambda_j \hat{H}_3^{ij}(s_1, s_2, s_3),
\end{aligned}$$

where the superscript notation is similar to Eq. (5.6). The above problem can be written as a classical quadratic optimization problem. We introduce the following notation: $\mathcal{A} = \hat{H}_3^{(ij)}(s_1, s_2, s_3)$, $\mathcal{B} = \hat{H}_3^{(is)}(s_1, s_2, s_3) + \hat{H}_3^{(si)}(s_1, s_2, s_3)$, $\mathcal{C} = \hat{H}_3^{(ss)}(s_1, s_2, s_3) - \hat{H}_3(s_1, s_2, s_3)$. We reformulate the problem by denoting $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_m \end{bmatrix}^T$. The dimensions for a single measurement triplet (s_1, s_2, s_3) remain: $\mathcal{A} \in \mathbb{R}^{n \times n}$, $\mathcal{B} \in \mathbb{R}^{1 \times n}$, $\mathcal{C} \in \mathbb{R}$.

$$\boldsymbol{\lambda}^T \mathcal{A} \boldsymbol{\lambda} + \mathcal{B}\boldsymbol{\lambda} + \mathcal{C} = 0. \tag{5.12}$$

We can rewrite the above vector equation in a more convenient format after vectorizing $\mathcal{A}$ as:

$$\mathtt{vec}(\mathcal{A})(\boldsymbol{\lambda} \otimes \boldsymbol{\lambda}) + \mathcal{B}\boldsymbol{\lambda} + \mathcal{C} = 0. \tag{5.13}$$

[1] The vectorization is row-wise, $vec(\mathbf{Q}) = \begin{bmatrix} \mathbf{Q}(1, 1:r^2) & \cdots & \mathbf{Q}(r, 1:r^2) \end{bmatrix}^T \in \mathbb{R}^{r^3 \times 1}$.

To enforce interpolation from the 3rd kernel, we equate

$$\underbrace{H_3\left(s_1^{(k)}, s_2^{(k)}, s_3^{(k)}\right)}_{\text{k: measurements}} = \hat{H}_3\left(s_1^{(k)}, s_2^{(k)}, s_3^{(k)}\right). \tag{5.14}$$

Further, by adding k measurements, we result to:

$$\underbrace{\begin{bmatrix} \mathtt{vec}(\mathcal{A}_1) \\ \mathtt{vec}(\mathcal{A}_2) \\ \vdots \\ \mathtt{vec}(\mathcal{A}_k) \end{bmatrix}}_{\mathbf{W}:\ (k\times m^2)} (\boldsymbol{\lambda}\otimes\boldsymbol{\lambda}) + \underbrace{\begin{bmatrix} \mathcal{B}_1 \\ \mathcal{B}_2 \\ \vdots \\ \mathcal{B}_k \end{bmatrix}}_{\mathbf{Z}:\ (k\times m)} \boldsymbol{\lambda} + \underbrace{\begin{bmatrix} \mathcal{C}_1 \\ \mathcal{C}_2 \\ \vdots \\ \mathcal{C}_k \end{bmatrix}}_{\mathbf{S}:\ (k\times 1)} = \mathbf{0}. \tag{5.15}$$

The above mapping can be written by denoting $\mathbf{F}(\cdot):\mathbb{R}^m \to \mathbb{R}^k$ as $\mathbf{F}(\boldsymbol{\lambda}) = \mathbf{0}$ where:

$$\mathbf{F}(\boldsymbol{\lambda}) = \mathbf{W}(\boldsymbol{\lambda}\otimes\boldsymbol{\lambda}) + \mathbf{Z}\boldsymbol{\lambda} + \mathbf{S},\ \boldsymbol{\lambda}\in\mathbb{R}^m. \tag{5.16}$$

The derivative (Jacobian) w.r.t the real vector $\boldsymbol{\lambda}$ is:

$$\mathbf{J}(\boldsymbol{\lambda}) = \mathbf{F}'(\boldsymbol{\lambda}) = \mathbf{W}(\boldsymbol{\lambda}\otimes\mathbf{I} + \mathbf{I}\otimes\boldsymbol{\lambda}) + \mathbf{Z}. \tag{5.17}$$

We seek the solution of Eq. (5.15); thus, by introducing the Newton iterative procedure (fixed point iterations), we can conclude in the following scheme where an initial seed $\boldsymbol{\lambda}_0$ can result to $\mathcal{F}(\boldsymbol{\lambda}_{n+1}) \to 0$ as $n\to\infty$. The iterations are described next:

$$\boldsymbol{\lambda}_{n+1} = \boldsymbol{\lambda}_n - \mathbf{J}^{-1}(\boldsymbol{\lambda}_n)\mathbf{F}(\boldsymbol{\lambda}_n). \tag{5.18}$$

Finally, upon Newton's method convergence, we obtain the vector $\boldsymbol{\lambda}^*$, $(\mathbf{F}(\boldsymbol{\lambda}^*)\approx\mathbf{0})$ from Algorithm 5.1, which will lead to a better estimation of $\mathbf{Q}$ that explains, in addition, the measurements from the 3rd kernel. We notice in many situations that the error between the reduced and original systems improves significantly when the residual γ of Newton's method remains small. Moreover, in many cases, identifying the original operator $\mathbf{Q}$ is possible, as we illustrate in the following example with the Lorenz attractor model.

Algorithm 5.1: Solution algorithm of the quadratic vector equation.

```
Define: $\mathbf{W}\in\mathbb{R}^{k\times m^2}$, $\mathbf{Z}\in\mathbb{R}^{k\times m}$, $\mathbf{S}\in\mathbb{R}^{k\times 1}$ and the hyperparameters $\eta$, $\gamma_0$.
Choose an initial random seed: $\boldsymbol{\lambda}\in\mathbb{R}^m$.
while $\gamma > \gamma_0$ do
    Compute $\mathbf{F}(\boldsymbol{\lambda})$ from Eq. (5.16) and $\mathbf{J}(\boldsymbol{\lambda})$ from Eq. (5.17).
    Update $\boldsymbol{\lambda} \leftarrow (\boldsymbol{\lambda} - \mathbf{J}^{\#}\mathbf{F}(\boldsymbol{\lambda}))$, # is: " $-1$" or the Moore-Penrose pseudo-inverse
    (threshold $\eta$).
    Compute the residue $\|\mathbf{F}(\boldsymbol{\lambda})\| = \gamma$.
end while
return $\boldsymbol{\lambda}$
```

5.3.4. Algorithm for quadratic modeling from i/o time-domain data

Here, we present a concise algorithm summarizing the procedure for constructing quadratic state space models from harmonic data (samples of the symmetric kernels $H_1,\ H_2,\ H_3$). Measuring (symmetric) Volterra kernels is no longer a new topic. However, although previously addressed in [37, 104, 154], it remains a non-trivial task. The main difficulty has to do with the fact that it is hard to separate commensurate frequencies. In other words, each of the propagating harmonics consists of a series of kernels and, therefore, evaluating the symmetric GFRFs requires kernel separation with an amplitude shifting [104, 154]. Towards this aim, X-parameters in [150], and the references within, represent a direct generalization of the classical S-parameters (for linear dynamics) to the nonlinear case. With this agile machinery, estimations of the higher Volterra kernel can be brought in an actual engineering setup as in [154], and a quadratic state-space surrogate model can be inferred from the proposed method. The following algorithm can use such information (from the X-parameters) to construct quadratic interpretable models.

Algorithm 5.2: Quadratic modeling from time-domain data

```
Input: # Measurements of the symmetric GFRFs $H_1,\ H_2,\ H_3$.
Output: The quadratic model of order $r$ with operators ($\hat{\mathbf{A}},\ \hat{\mathbf{Q}},\ \hat{\mathbf{B}},\ \hat{\mathbf{C}}$).
  Define a truncation order $r$ with SVD from the Loewner matrix $\mathbb{L}$.
  Realize the minimal linear subsystem ($\hat{\mathbf{A}},\ \hat{\mathbf{B}},\ \hat{\mathbf{C}}$) of order $r$.
  Estimate the $\hat{\mathbf{Q}}_s \in \mathbb{R}^{r\times r^2}$ from Eq. (5.9) by minimizing the 2−norm error.
  Update $\hat{\mathbf{Q}} \in \mathbb{R}^{r\times r^2}$ from Eq. (5.10) after solving Eq. (5.18) with Algorithm 5.1.
```

5.3.5. Quadratic state-space systems with multiple equilibrium points

Quadratic systems can bifurcate to different equilibrium points that operate locally. Thus, when measuring, multi-operational points can be revealed. To illustrate this phenomenon mathematically, we write the quadratic system Eq. (5.1) after shifting it with the non-zero equilibrium state $\mathbf{x}_e$. We denote the new state variable $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_e$, and it remains

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{Q}(\mathbf{x}(t)\otimes\mathbf{x}(t)) + \mathbf{B}u(t) \Rightarrow\\
\dot{\tilde{\mathbf{x}}}(t) &= \mathbf{A}(\tilde{\mathbf{x}}(t)+\mathbf{x}_e) + \mathbf{Q}\left((\tilde{\mathbf{x}}(t)+\mathbf{x}_e)\otimes(\tilde{\mathbf{x}}(t)+\mathbf{x}_e)\right) + \mathbf{B}u(t) \Rightarrow\\
\dot{\tilde{\mathbf{x}}}(t) &= \mathbf{A}\tilde{\mathbf{x}}(t) + 2\mathbf{Q}(\mathbf{x}_e\otimes\tilde{\mathbf{x}}(t)) + \mathbf{Q}(\tilde{\mathbf{x}}(t)\otimes\tilde{\mathbf{x}}(t)) + \mathbf{A}\mathbf{x}_e + \mathbf{Q}(\mathbf{x}_e\otimes\mathbf{x}_e) + \mathbf{B}u(t) \Rightarrow\\
\dot{\tilde{\mathbf{x}}}(t) &= \underbrace{(\mathbf{A}+2\mathbf{Q}(\mathbf{x}_e\otimes\mathbf{I}))}_{\tilde{\mathbf{A}}}\tilde{\mathbf{x}}(t) + \mathbf{Q}(\tilde{\mathbf{x}}(t)\otimes\tilde{\mathbf{x}}(t)) + \underbrace{\mathbf{A}\mathbf{x}_e + \mathbf{Q}(\mathbf{x}_e\otimes\mathbf{x}_e)}_{\tilde{\mathbf{L}}} + \mathbf{B}u(t).
\end{aligned} \tag{5.19}$$

Note that $\mathbf{L} := \mathbf{A}\mathbf{x}_e + \mathbf{Q}(\mathbf{x}_e\otimes\mathbf{x}_e) = \mathbf{0}$, and should remain zero as in the absence of the controller $u(t)$, and with zero initial conditions, e.g., $\mathbf{x}_0 = \mathbf{0}$, there is no energy in the system to dissipate. We do not address situations with a limit circle, e.g., systems with purely imaginary eigenvalues, for which such systems describe self-sustained dynamics. As a result, the quadratic system that we measure after reaching the equilibrium state

$\mathbf{x}_e$ is the following:

$$\begin{cases}\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}\tilde{\mathbf{x}}(t) + \mathbf{Q}(\tilde{\mathbf{x}}(t)\otimes\tilde{\mathbf{x}}(t)) + \mathbf{B}u(t),\\ y(t) = \mathbf{C}\tilde{\mathbf{x}}(t) + \mathbf{C}\mathbf{x}_e.\end{cases} \tag{5.20}$$

Remark 5.1 (Invariant operators under bifurcations):
The system in Eq. (5.20) suggests that around the new equilibrium state point $\mathbf{x}_e$, the operators $(\mathbf{Q},\ \mathbf{B},\ \mathbf{C})$ stay invariant, and only the linear operator changes to $\tilde{\mathbf{A}} = \mathbf{A} + 2\mathbf{Q}(\mathbf{x}_e \otimes \mathbf{I})$, along with the DC[2] term $\mathbf{C}\mathbf{x}_e$. Therefore, for the multiple equilibrium case, these local systems contain the same invariant information w.r.t. the operators $(\mathbf{Q},\ \mathbf{B},\ \mathbf{C})$ except the linear operator plus a translation. In other words, the generalized Markov parameters of the system that contain only the operators $(\mathbf{Q},\ \mathbf{B},\ \mathbf{C})$ are the same around any arbitrary equilibrium $\mathbf{x}_e$ to which the original system bifurcates and any arbitrary coordinate system. ◊

Two equilibrium points case: Let assume that the original quadratic model has bifurcated to the two different equilibrium points $\hat{\mathbf{x}}_e^{(1)}$, $\breve{\mathbf{x}}_e^{(2)}$ and in the different coordinates denoted $(\hat{\mathbf{x}},\ \breve{\mathbf{x}})$, that explain the dynamical behavior locally. We can write

$$\begin{aligned}&\begin{cases}\dot{\hat{\mathbf{x}}}_1(t) = \hat{\mathbf{A}}_1\hat{\mathbf{x}}_1(t) + \hat{\mathbf{Q}}_1(\hat{\mathbf{x}}_1(t)\otimes\hat{\mathbf{x}}_1(t)) + \hat{\mathbf{B}}_1u(t),\\ y_1(t) = \hat{\mathbf{C}}_1\hat{\mathbf{x}}_1(t),\ \hat{\mathbf{x}}_1(0) = \mathbf{0}.\end{cases},\\ &\begin{cases}\dot{\breve{\mathbf{x}}}_2(t) = \breve{\mathbf{A}}_2\breve{\mathbf{x}}_2(t) + \breve{\mathbf{Q}}_2(\breve{\mathbf{x}}_2(t)\otimes\breve{\mathbf{x}}_2(t)) + \breve{\mathbf{B}}_2u(t),\\ y_2(t) = \breve{\mathbf{C}}_2\breve{\mathbf{x}}_2(t),\ \breve{\mathbf{x}}_2(0) = \mathbf{0}.\end{cases}\end{aligned} \tag{5.21}$$

Some properties:

- For the first system in Eq. (5.21) holds

$$\underbrace{\hat{\mathbf{A}}_1}_{\text{local}} = \underbrace{\mathbf{A}_1}_{\text{global}} + 2\hat{\mathbf{Q}}_1(\hat{\mathbf{x}}_e^{(1)}\otimes\mathbf{I}) \tag{5.22}$$

- For the second system in Eq. (5.21) holds

$$\underbrace{\breve{\mathbf{A}}_2}_{\text{local}} = \underbrace{\mathbf{A}_2}_{\text{global}} + 2\breve{\mathbf{Q}}_2(\breve{\mathbf{x}}_e^{(2)}\otimes\mathbf{I}) \tag{5.23}$$

Remark 5.2 (Markov parameters):
The Markov parameters involving the quadratic and input-output operators are the same. $\hat{\mathbf{C}}_1\hat{\mathbf{B}}_1 = \breve{\mathbf{C}}_2\breve{\mathbf{B}}_2$, and $\hat{\mathbf{C}}_1\hat{\mathbf{Q}}_1(\hat{\mathbf{B}}_1\otimes\hat{\mathbf{B}}_1) = \breve{\mathbf{C}}_2\breve{\mathbf{Q}}_2(\breve{\mathbf{B}}_2\otimes\breve{\mathbf{B}}_2)$. ◊

[2] DC: direct current in electrical engineering, which describes the non-periodic term (zero frequency) in the power spectrum.

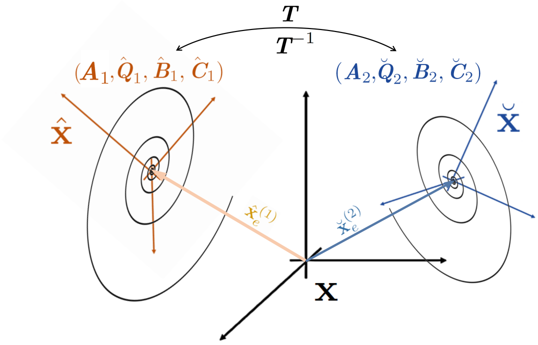

$\boldsymbol{T}$
$\boldsymbol{T}^{-1}$
$(\boldsymbol{A}_1, \hat{\boldsymbol{Q}}_1, \hat{\boldsymbol{B}}_1, \hat{\boldsymbol{C}}_1)$
$(\boldsymbol{A}_2, \breve{\boldsymbol{Q}}_2, \breve{\boldsymbol{B}}_2, \breve{\boldsymbol{C}}_2)$
$\hat{\mathbf{X}}$
$\breve{\mathbf{X}}$
$\hat{\mathbf{x}}_e^{(1)}$
$\breve{\mathbf{x}}_e^{(2)}$
$\mathbf{X}$

Figure 5.1.: Lemma 5.3 through a schematic.

According to Remark 5.2, invariant information from the original system is encoded in both systems Eq. (5.21). Therefore, a similarity transformation $\mathbf{T}$ exists, which aligns the two systems w.r.t the original operators.

Lemma 5.3 (Quadratic systems alignement):
There exists a transformation matrix $\mathbf{T}$ such that the two triplets of operators given by $(\hat{\mathbf{Q}}_1, \hat{\mathbf{B}}_1, \hat{\mathbf{C}}_1)$ and by $(\breve{\mathbf{Q}}_2, \breve{\mathbf{B}}_2, \breve{\mathbf{C}}_2)$ (resulting after a global model bifurcating to different equilibrium points) can be aligned simultaneously with the original operators but to different coordinates, geometrically in Fig. 5.1, and algebraically as

$$\begin{aligned}
\hat{\mathbf{Q}}_1 &= \mathbf{T}\breve{\mathbf{Q}}_2(\mathbf{T}^{-1} \otimes \mathbf{T}^{-1}), \\
\hat{\mathbf{B}}_1 &= \mathbf{T}^{-1}\breve{\mathbf{B}}_2 \Leftrightarrow \mathbf{T}\hat{\mathbf{B}}_1 = \breve{\mathbf{B}}_2 \\
\hat{\mathbf{C}}_1 &= \mathbf{T}\breve{\mathbf{C}}_2, \\
\mathbf{A}_1 &= \mathbf{T}\mathbf{A}_2\mathbf{T}^{-1}.
\end{aligned} \tag{5.24}$$

$\Diamond$

One way to compute the transformation matrix $\mathbf{T}$ is by solving the first three equations in system Eq. (5.24). The above problem involves a quadratic matrix equation that can be iteratively solved using Newton iterations. Moreover, the linear constraints help the regularization of the Newton iterations not to converge at the zero solution. We analytically derive the iterative Newton scheme over the Fréchet derivative in what follows (Section 5.3.6) to seek such a formal solution.

5.3.6. Solution of the constrained quadratic matrix equation

The analysis starts with the quadratic matrix equation. Thus, we define the following operator: $\mathcal{F} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ with $\mathcal{F}(\mathbf{X}) := \mathbf{X}\mathbf{U} - \mathbf{Q}\,(\mathbf{X} \otimes \mathbf{X})$. For known $\mathbf{U},\ \mathbf{Q} \in \mathbb{R}^{n\times n^2}$, we seek $\mathbf{0} \neq \mathbf{X} \in \mathbb{R}^{n\times n}$ such that $\mathcal{F}(\mathbf{X}) = \mathbf{0}$. Moreover, $\mathbf{X}$ should be invertible ($\exists\ \mathbf{X}^{-1}$). The idea is to differentiate w.r.t. the Fréchet derivative and solve a linear matrix equation for every Newton step similar to the Newton-Kleinmann algorithm for the solution of the Ricatti matrix equation [112]. Therefore, we introduce a small perturbation to the matrix

$\mathbf{X}$ with $\mathbf{N} \in \mathbb{R}^{n\times n}$ and with h a small real number. We define

$$(\mathcal{F}'(\mathbf{X}))(\mathbf{N}) = \lim_{h\to 0} \frac{1}{h}\left(\mathcal{F}(\mathbf{X}+h\mathbf{N}) - \mathcal{F}(\mathbf{X})\right) = \mathbf{N}\mathbf{U} - \mathbf{Q}(\mathbf{X}\otimes\mathbf{N} + \mathbf{N}\otimes\mathbf{X}). \tag{5.25}$$

Since $\mathbf{Q}$ is symmetric, we can write equivalently

$$(\mathcal{F}'(\mathbf{X}))(\mathbf{N}) = \mathbf{N}\mathbf{U} - 2\mathbf{Q}(\mathbf{X}\otimes\mathbf{N}). \tag{5.26}$$

The Newton iteration is given by

$$(\mathcal{F}'(\mathbf{X}_{j-1}))(\mathbf{N}_{j-1}) = -\mathcal{F}(\mathbf{X}_{j-1}),\ \mathbf{X}_j = \mathbf{X}_{j-1} + \mathbf{N}_{j-1}. \tag{5.27}$$

We compute

$$\begin{aligned} \mathbf{N}_{j-1}\mathbf{U} - 2\mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{N}_{j-1}) &= -\mathbf{X}_{j-1}\mathbf{U} + \mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_{j-1}) \Rightarrow \\ (\mathbf{X}_j - \mathbf{X}_{j-1})\mathbf{U} - 2\mathbf{Q}(\mathbf{X}_{j-1}\otimes(\mathbf{X}_j - \mathbf{X}_{j-1})) &= -\mathbf{X}_{j-1}\mathbf{U} + \mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_{j-1}) \Rightarrow \\ \mathbf{X}_j\mathbf{U} - 2\mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_j) + 2\mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_{j-1}) &= \mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_{j-1}) \end{aligned} \tag{5.28}$$

which results to the following linear matrix equation Eq. (5.29) w.r.t the forward step solution $\mathbf{X}_j$:

$$\mathbf{X}_j\mathbf{U} - 2\mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_j) + \mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_{j-1}) = 0. \tag{5.29}$$

Remark 5.4:
In Eq. (5.29), it is to be observed that at step j, the matrix equation is linear in $\mathbf{X}_j$, provided that $\mathbf{X}_{j-1}$ is explicitly known, which is to be assumed (from the Newton iteration). ◊

Remark 5.5:
The equation (5.29) is linear in the variable $\mathbf{X}_j \in \mathbb{R}^{n\times n}$; since $\mathbf{U}, \mathbf{Q} \in \mathbb{R}^{n\times n^2}$, there are n^3 linear scalar equations to solve, and only n^2 unknowns. Hence, we are facing an over-determined linear system of equations with a possibly non-empty null space. ◊

In what follows, we show how to isolate the $\mathbf{X}_j$ term from the rest and how to rewrite this equation more conventionally. More specifically, based on the previous remark, we show that equation (5.29) can equivalently be written as n classical Sylvester equations, each characterized by n^2 scalar equations in n^2 unknowns. From (5.29), it follows that

$$\begin{aligned} \mathbf{X}_j\mathbf{U} - 2\mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_j) + \mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_{j-1}) = 0 \Rightarrow \\ \mathbf{X}_j\mathbf{U} - \underbrace{2\mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{I}_n)}_{:=\mathbf{V}_{j-1}}(\mathbf{I}_n\otimes\mathbf{X}_j) = \underbrace{-\mathbf{Q}(\mathbf{X}_{j-1}\otimes\mathbf{X}_{j-1})}_{:=\mathbf{Z}_{j-1}} \Rightarrow \end{aligned} \tag{5.30}$$

$$\mathbf{X}_j \underbrace{\begin{bmatrix}\mathbf{U}^{(1)} & \cdots\mathbf{U}^{(n)}\end{bmatrix}}_{\mathbf{U}} - \underbrace{\begin{bmatrix}\mathbf{V}_{j-1}^{(1)} & \cdots\mathbf{V}_{j-1}^{(n)}\end{bmatrix}}_{\mathbf{V}_{j-1}} \begin{bmatrix}\mathbf{X}_j & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{X}_j & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X}_j\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{Z}_{j-1}^{(1)} & \mathbf{Z}_{j-1}^{(2)} & \cdots\mathbf{Z}_{j-1}^{(n)}\end{bmatrix}}_{\mathbf{Z}_{j-1}}. \tag{5.31}$$

Above, we have that $\mathbf{U}^{(k)}, \mathbf{V}_{j-1}^{(k)}, \mathbf{Z}_{j-1}^{(k)}$ are known $n \times n$ real-valued matrices at step j, for all $1 \leq k \leq n$. These are the building blocks of the following matrices:

$$\mathbf{V}_{j-1} := 2\mathbf{Q}(\mathbf{X}_{j-1} \otimes \mathbf{I}_n) \in \mathbb{R}^{n \times n^2}, \quad \mathbf{Z}_{j-1} := -\mathbf{Q}(\mathbf{X}_{j-1} \otimes \mathbf{X}_{j-1}) \in \mathbb{R}^{n \times n^2}. \tag{5.32}$$

We can hence write this equation equivalently as follows:

$$\begin{bmatrix}\mathbf{X}_j\mathbf{U}^{(1)} & \cdots \mathbf{X}_j\mathbf{U}^{(n)}\end{bmatrix} - \begin{bmatrix}\mathbf{V}_{j-1}^{(1)} & \cdots \mathbf{V}_{j-1}^{(n)}\mathbf{X}_j\end{bmatrix} = \begin{bmatrix}\mathbf{Z}_{j-1}^{(1)} & \cdots \mathbf{Z}_{j-1}^{(n)}\end{bmatrix}. \tag{5.33}$$

Then, for all $1 \leq k \leq n$, solving (5.29) boils down to solving n (linear) Sylvester equations as:

$$\mathbf{X}_j\mathbf{U}^{(k)} - \mathbf{V}_{j-1}^{(k)}\mathbf{X}_j = \mathbf{Z}_{j-1}^{(k)}. \tag{5.34}$$

The solution $\mathbf{X}_j \in \mathbb{R}^{n \times n}$, after vectorization, becomes $\text{vec}(\mathbf{X}_j) \in \mathbb{R}^{n^2 \times 1}$. Putting together the n Sylvester equations in vectorized form by using the identity $\text{vec}(\mathbf{TOR}) = (\mathbf{R}^T \otimes \mathbf{T})\text{vec}(\mathbf{O})$, will yield the following system of n^3 scalar equations in n^2 unknowns:

$$\underbrace{\begin{bmatrix}\left(\mathbf{U}^{(1)}\right)^T \otimes \mathbf{I}_n - \mathbf{I}_n \otimes \mathbf{V}_{j-1}^{(1)} \\ \left(\mathbf{U}^{(2)}\right)^T \otimes \mathbf{I}_n - \mathbf{I}_n \otimes \mathbf{V}_{j-1}^{(2)} \\ \vdots \\ \left(\mathbf{U}^{(n)}\right)^T \otimes \mathbf{I}_n - \mathbf{I}_n \otimes \mathbf{V}_{j-1}^{(n)}\end{bmatrix}}_{\in \mathbb{R}^{n^3 \times n^2}} \text{vec}(\mathbf{X}_j) = \underbrace{\begin{bmatrix}\text{vec}(\mathbf{Z}_{j-1}^{(1)}) \\ \text{vec}(\mathbf{Z}_{j-1}^{(2)}) \\ \vdots \\ \text{vec}(\mathbf{Z}_{j-1}^{(n)})\end{bmatrix}}_{\in \mathbb{R}^{n^3 \times 1}} \tag{5.35}$$

For low values of n, such a procedure is indeed feasible. However, for moderate to large values of n, i.e., $n > 50$ or so, it is quite challenging or even impossible to find the next value $\mathbf{X}_j$ using explicitly forming the $n^3 \times n^2$ matrix in (5.35). In what follows, we are concerned with low-order systems as we emphasize quadratic identification in a reduced-order sense.

Lemma 5.6 (Linearized system of Sylvester equations):
The square matrix $\mathbf{T}^{-1}$ that aligns the operators $(\hat{\mathbf{Q}}_1, \hat{\mathbf{B}}_1, \hat{\mathbf{C}}_1)$ and $(\breve{\mathbf{Q}}_2, \breve{\mathbf{B}}_2, \breve{\mathbf{C}}_2)$ from Lemma 5.3, can be computed, upon Newton's method convergence Algorithm 5.3, as the iterative solution of the following constrained linear system of equations Eq. (5.36) with $\mathbf{T}^{-1} := \lim_{j\to\infty} \mathbf{X}_j$ that gives $\mathcal{F}(\mathbf{T}^{-1}) \approx \mathbf{0}$.

$$\underbrace{\begin{bmatrix}\left(\mathbf{U}^{(1)}\right)^T \otimes \mathbf{I}_n - \mathbf{I}_n \otimes \mathbf{V}_{j-1}^{(1)} \\ \left(\mathbf{U}^{(2)}\right)^T \otimes \mathbf{I}_n - \mathbf{I}_n \otimes \mathbf{V}_{j-1}^{(2)} \\ \vdots \\ \left(\mathbf{U}^{(n)}\right)^T \otimes \mathbf{I}_n - \mathbf{I}_n \otimes \mathbf{V}_{j-1}^{(n)} \\ \hat{\mathbf{B}}_1^T \otimes \mathbf{I}_n \\ \mathbf{I}_n \otimes \breve{\mathbf{C}}_2\end{bmatrix}}_{\in \mathbb{R}^{(n^3+2n) \times n^2}} \underbrace{\text{vec}(\mathbf{X}_j)}_{\in \mathbb{R}^{n^2 \times 1}} = \underbrace{\begin{bmatrix}\text{vec}(\mathbf{Z}_{j-1}^{(1)}) \\ \text{vec}(\mathbf{Z}_{j-1}^{(2)}) \\ \vdots \\ \underbrace{\text{vec}(\mathbf{Z}_{j-1}^{(n)})}_{\in \mathbb{R}^{n^2 \times 1}} \\ \breve{\mathbf{B}}_2 \\ \hat{\mathbf{C}}_1^T\end{bmatrix}}_{\in \mathbb{R}^{(n^3+2n) \times 1}} \tag{5.36}$$

◊

Algorithm 5.3: Solution algorithm for the constrained quadratic matrix equation

```
Seek: $\mathbf{X}$ s.t. $\mathcal{F}(\mathbf{X}) := \mathbf{XU} - \mathbf{Q}(\mathbf{X} \otimes \mathbf{X}) = \mathbf{0}$ and satisfies the constrains (two last
    rows) in Eq. (5.36).
Choose an initial random seed: $\mathbf{X}_{j=0} \in \mathbb{R}^{n \times n}$.
while $\gamma > \gamma_0$ do
    Update: $j \leftarrow j + 1$.
    Compute $\mathbf{X}_j$ by solving the linear system of equations Eq. (5.36).
    Compute the residue $\|\mathcal{F}(\mathbf{X}_j)\| = \gamma$.
end while
return  $\mathbf{X}$
```

With the solution $\mathbf{T}^{-1}$ from Algorithm 5.3, we can align the "hatted" and "breved" systems to the same coordinates. We can further write after combining equations Eq. (5.22) and Eq. (5.23), the following system with unknowns the equilibrium state points $\hat{\mathbf{x}}_2^{(1)}$, $\breve{\mathbf{x}}_2^{(2)}$. Combining equations Eqs. (5.22) and (5.23) after multiplication with the transformation matrix $\mathbf{T}$ from the left and with $\mathbf{T}^{-1}$ from the right, we have

$$\begin{aligned}
\breve{\mathbf{A}}_2 &= \mathbf{A}_2 + 2\breve{\mathbf{Q}}_2(\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{I}) \Rightarrow \\
\mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= \mathbf{T}\mathbf{A}_2\mathbf{T}^{-1} + 2\mathbf{T}\breve{\mathbf{Q}}_2(\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{I})\mathbf{T}^{-1} \Rightarrow \\
\mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= \mathbf{A}_1 + 2\mathbf{T}\breve{\mathbf{Q}}_2(\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{T}^{-1}) \Rightarrow \\
\mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= \mathbf{A}_1 + 2\mathbf{T}\breve{\mathbf{Q}}_2(\mathbf{T}^{-1} \otimes \mathbf{T}^{-1})(\mathbf{T}^{-1} \otimes \mathbf{T}^{-1})^{-1}(\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{T}^{-1}) \\
\mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= \mathbf{A}_1 + 2\hat{\mathbf{Q}}_1(\mathbf{T}^{-1} \otimes \mathbf{T}^{-1})^{-1}(\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{T}^{-1}) \Rightarrow \\
\mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= \mathbf{A}_1 + 2\hat{\mathbf{Q}}_1(\mathbf{T} \otimes \mathbf{T})(\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{T}^{-1}) \Rightarrow \\
\mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= \mathbf{A}_1 + 2\hat{\mathbf{Q}}_1(\mathbf{T}\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{I}) \Rightarrow \\
\mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= \hat{\mathbf{A}}_1 - 2\hat{\mathbf{Q}}_1(\hat{\mathbf{x}}_e^{(1)} \otimes \mathbf{I}) + 2\hat{\mathbf{Q}}_1(\mathbf{T}\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{I}) \Rightarrow \\
\mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= \hat{\mathbf{A}}_1 - 2\hat{\mathbf{Q}}_1\left(\hat{\mathbf{x}}_e^{(1)} - \mathbf{T}\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{I}\right) \Rightarrow \\
\hat{\mathbf{A}}_1 - \mathbf{T}\breve{\mathbf{A}}_2\mathbf{T}^{-1} &= 2\hat{\mathbf{Q}}_1\left(\hat{\mathbf{x}}_e^{(1)} - \mathbf{T}\breve{\mathbf{x}}_e^{(2)} \otimes \mathbf{I}\right).
\end{aligned} \tag{5.37}$$

The above equation is not enough to define the unknown equilibrium vectors uniquely. Additional information comes from the direct current (DC) terms that can be measured from the power spectrum $\alpha_1,\ \alpha_2$. Therefore, we enforce from the Table 5.2

$$\hat{\mathbf{C}}_1\hat{\mathbf{x}}_e^{(1)} = \alpha_1, \quad \breve{\mathbf{C}}_2\breve{\mathbf{x}}_e^{(2)} \quad = \alpha_2, \tag{5.38}$$

Solving the coupled system with Eq. (5.37) and Eq. (5.38), we obtain infinite solution of the vectors $\hat{\mathbf{x}}_e^{(1)}$, $\breve{\mathbf{x}}_e^{(2)}$ as a non-empty null space of length p exists. Finally, each one of the systems at the equilibrium point satisfies $\mathbf{L} := \mathbf{A}\mathbf{x}_e + \mathbf{Q}(\mathbf{x}_e \otimes \mathbf{x}_e) = \mathbf{0}$. Therefore, working independently, at $\hat{\mathbf{x}}_e^{(1)}$, it holds $\hat{\mathbf{L}}\left(\hat{\mathbf{x}}_e^{(1)}\right) = \mathbf{0}$. The solution we estimate from Eq. (5.37) is not unique due to the rank deficiency. In particular, we have to solve for the two equilibrium points another two quadratic vector equations that enforce $\hat{\mathbf{L}}(\hat{\mathbf{x}}_e^{(1)})$

and $\breve{\mathbf{L}}(\breve{\mathbf{x}}_e^{(2)})$ equal zero. Therefore, we solve for the parametric solution

$$\begin{bmatrix} \mathbf{x}_e^{(1)} \\ \mathbf{x}_e^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{es}^{(1)} \\ \mathbf{x}_{es}^{(2)} \end{bmatrix} + \sum_{i=1}^{p} \lambda_i \begin{bmatrix} \mathbf{x}_{ei}^{(1)} \\ \mathbf{x}_{ei}^{(2)} \end{bmatrix}, \tag{5.39}$$

the following equation for every equilibrium point. Thus,

$$\begin{aligned} \hat{\mathbf{L}}\left(\hat{\mathbf{x}}_e^{(1)}\right) :&= \mathbf{A}_1\hat{\mathbf{x}}_e^{(1)} + \hat{\mathbf{Q}}_1(\hat{\mathbf{x}}_e^{(1)} \otimes \hat{\mathbf{x}}_e^{(1)}) = \hat{\mathbf{A}}_1\hat{\mathbf{x}}_e^{(1)} - \hat{\mathbf{Q}}_1\left(\hat{\mathbf{x}}_e^{(1)} \otimes \hat{\mathbf{x}}_e^{(1)}\right) = \\ &= \hat{\mathbf{A}}_1\left(\hat{\mathbf{x}}_{es}^{(1)} + \sum_{i=1}^{p}\lambda_i\mathbf{x}_{ei}^{(1)}\right) - \hat{\mathbf{Q}}_1\left(\hat{\mathbf{x}}_{es}^{(1)} + \sum_{i=1}^{p}\lambda_i\mathbf{x}_{ei}^{(1)}\right) \otimes \left(\hat{\mathbf{x}}_{es}^{(1)} + \sum_{i=1}^{p}\lambda_i\mathbf{x}_{ei}^{(1)}\right) = \\ &= \underbrace{\hat{\mathbf{A}}_1\hat{\mathbf{x}}_{es}^{(1)} - \hat{\mathbf{Q}}_1\left(\hat{\mathbf{x}}_{es}^{(1)} \otimes \hat{\mathbf{x}}_{es}^{(1)}\right)}_{\mathbf{Z}} + \sum_{i=1}^{p}\lambda_i \underbrace{\left(\hat{\mathbf{A}}_1\mathbf{x}_{ei}^{(1)} - 2\hat{\mathbf{Q}}_1\mathbf{x}_{es}^{(1)} \otimes \mathbf{x}_{ei}^{(1)}\right)}_{\mathbf{Y}} - \\ &- \sum_{i=1}^{p}\sum_{j=1}^{p}\lambda_i\lambda_j \underbrace{\hat{\mathbf{Q}}_1\left(\mathbf{x}_{ei}^{(1)} \otimes \mathbf{x}_{ej}^{(1)}\right)}_{\mathbf{W}} = \mathbf{0}. \end{aligned} \tag{5.40}$$

Therefore, after enforcing the $\mathbf{L}$ vector to be zero at each equilibrium point and for both systems, we solve a system of coupled quadratic vector equations.

$$\begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix} (\boldsymbol{\lambda} \otimes \boldsymbol{\lambda}) + \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} \boldsymbol{\lambda} + \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{bmatrix} = \mathbf{0}. \tag{5.41}$$

Solving for $\boldsymbol{\lambda}$ with the same developed Algorithm 5.1, we can detect uniquely the equilibrium state vectors $\hat{\mathbf{x}}_e^{(1)}$, $\breve{\mathbf{x}}_e^{(2)}$. Finally, we can identify the initial system that contains the original operators, e.g., $\mathbf{A}_1$. In particular, the identified system with operators $(\mathbf{A}_1,\ \hat{\mathbf{Q}}_1,\ \hat{\mathbf{B}}_1,\ \hat{\mathbf{C}}_1)$ is an equivalent modulo with the original $(\mathbf{A},\ \mathbf{Q},\ \mathbf{B},\ \mathbf{C})$ and a similarity transformation $\tilde{\mathbf{T}} \in \mathbb{R}^{n\times n}$ exists that aligns the two systems.

5.4. Numerical results

We test the new method for different cases of identifying the Lorenz attractor where the Burgers' equation model illustrates the reduction performance.

5.4.1. Identification of the forced Lorenz system

We consider the canonical model for chaotic dynamics, the Lorenz system [120], and we add a control-input $u(t)$ in the 1st and 3rd states. The following state-space form describes the quadratic control system:

$$\begin{cases} \dot{x}(t) = -\sigma x(t) + \sigma y(t) + u(t), \\ \dot{y}(t) = \rho x(t) - y(t) - x(t)z(t), \\ \dot{z}(t) = -\beta z(t) + x(t)y(t) + u(t), \end{cases} \tag{5.42}$$

where zero initial condition are assumed e.g., $(x(0),\ y(0),\ z(0)) = (0,\ 0,\ 0)$, and the operators are:

$$\mathbf{A} = \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{bmatrix},\ \mathbf{B} = \mathbf{C}^T = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},\ \mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.43}$$

With input $u(t)$, we choose to observe the linear combination of the 1st and 3rd states; thus, the output is $x(t)+z(t)$. The above quadratic system Eq. (5.43) gives rise to chaotic dynamics for different choices of the parameters $(\sigma,\ \rho,\ \beta)$. This study aims to identify the Lorenz system from i/o time domain data under harmonic excitation. We choose $\sigma = 10,\ \beta = 8/3$, and for the parameter ρ, we investigate two cases 1,2 and comment on case 3.

1. $\rho = 0.5$, where the linear subsystem is stable, and the Lorenz attractor has the unique zero equilibrium.

2. $\rho = 20$, where the linear subsystem is unstable, and the Lorenz system has two different steady-states with two non-zero stable equilibrium points.

3. $\rho = 28$, where the linear subsystem is unstable, but the Lorenz system is chaotic (steady-state unreachable) with two non-trivial attractors.

Case 1 - $\rho = 0.5$. Exciting the Lorenz system Eq. (5.43) with multi-harmonic inputs, e.g., $u(t) = a_1 e^{s_1 t} + a_2 e^{s_2 t} + a_3 e^{s_3 t}$, where $j = \sqrt{-1}$ the imaginary unit, and $s_1 = j\omega_1,\ s_2 = j\omega_2,\ s_3 = j\omega_3$, after reaching the steady-state profile, measurements of the GFRFs can be achieved, e.g., with X-parameters. The data assimilation process is repetitive, and for the real input case, e.g., $u(t) = \sum_{i=1}^{n} cos(\omega_i t)$, kernel separation should be addressed in a similar way as in [104, 37, 36]. Therefore, samples of the first three GFRFs over the following frequency grids can be obtained from a physical measurement setup after processing the time-domain evolution of the potentially unknown system.

- We take 50 logarithmic distributed measurements $\omega_i,\ i = 1, \ldots, 50$, from $[10^{-2}, 10^{2}]$. Therefore, 50 pairs of measurements $\{j\omega_i,\ H_1(j\omega_i)\},\ i = 1, \ldots, 50$ are collected. Using the Loewner framework Section 3.2, the order $r = 3$ of the linear minimal subsystem can be identified from the singular value decay Fig. 5.2(left), and a linear realization can be constructed:

$$\hat{\mathbf{A}} = \begin{bmatrix} 16.96 & 4.171 & -4.638 \\ 11.32 & 4.408 & -4.287 \\ 135.2 & 29.95 & -35.03 \end{bmatrix},\ \hat{\mathbf{B}} = \begin{bmatrix} -3.047 \\ -1.824 \\ -24.97 \end{bmatrix},\ \hat{\mathbf{C}}^T = \begin{bmatrix} 2.542 \\ 0.08718 \\ -0.3968 \end{bmatrix}. \tag{5.44}$$

  The coordinate system is different from the original, but the system's invariant quantities are the same, e.g., the 1st transfer function H_1, or Markov parameters, e.g., $\mathbf{CAB} = \hat{\mathbf{C}}\hat{\mathbf{A}}\hat{\mathbf{B}} = -12.6667$. The eigenvalues are: $\texttt{eig}(\mathbf{A}) = \texttt{eig}(\hat{\mathbf{A}}) = \begin{pmatrix} -10.52 & -0.4751 & -2.667 \end{pmatrix}$.

- We take 10 logarithmic distributed measurements from a squared grid $[10^{-2}, 10^{2}]^2$ in each dimension[3], and 100 pairs of measurements $\{(j\omega_1^k,\ j\omega_2^k),\ H_2(j\omega_1^k,\ j\omega_2^k)\}$

[3] Cartesian product: $[a, b]^2 = [a, b] \times [a, b]$ for $a < b$.

are collected. Solving the linear system Eq. (5.9) by minimizing the 2-norm (least-squares), we estimate $\hat{\mathbf{Q}}_s$ as

$$\hat{\mathbf{Q}}_s = \begin{bmatrix} -1.243 & 0.1493 & 0.2813 & 0.1493 & -0.01241 & 0.01805 & 0.2813 & 0.01805 & -0.05817 \\ 0.1416 & 0.8189 & 0.03877 & 0.8189 & -0.2759 & 0.03372 & 0.03877 & 0.03372 & -0.007071 \\ 0.4246 & 0.03703 & 0.8877 & 0.03703 & 0.1219 & 0.2355 & 0.8877 & 0.2355 & -0.2717 \end{bmatrix}.$$

- The rank of the least squares matrix in Eq. (5.9) is deficient $\texttt{rank}(\mathbf{M}) = 21 < 27 = 3^3$. Therefore, a parameterization is introduced as in Eq. (5.10). In this particular case, the dimension of the vector $\boldsymbol{\lambda}$ is six. As the proposed method is arbitrary in the number of measurements, we take 5 logarithmically distributed measurements from the cubic grid $[10^{-2}, 10^2]^3$ in each dimension; therefore, 125 pairs of measurements $\{(j\omega_1^k,\ j\omega_2^k,\ j\omega_3^k),\ H_3(j\omega_1^k,\ j\omega_2^k,\ j\omega_3^k)\}$ are collected. Solving the quadratic equation with Algorithm 5.1, and starting with different seeds of λ_0, as depicted on the right of Fig. 5.2, the parameter vector $\boldsymbol{\lambda} \in \mathbb{R}^6$ is obtained uniquely. Thus, the updated estimation of the quadratic operator $\hat{\mathbf{Q}} = \hat{\mathbf{Q}}_s + \sum_{i=1}^{r} \lambda \hat{\mathbf{Q}}_i$, with $\hat{\mathbf{Q}}_i$ the null space vectors is the following:

$$\hat{\mathbf{Q}} = \begin{bmatrix} -1.513 & -0.1403 & 0.2603 & -0.1403 & -0.006141 & 0.0241 & 0.2603 & 0.0241 & -0.04477 \\ 22.15 & 0.5147 & -3.506 & 0.5147 & 0.01186 & -0.08085 & -3.506 & -0.08085 & 0.5511 \\ -5.541 & -0.7411 & 0.9982 & -0.7411 & -0.03402 & 0.1284 & 0.9982 & 0.1284 & -0.1794 \end{bmatrix}.$$

Finally, using a coordinate transformation Appendix C, we prove that the resulting system is exactly the original.

$$\begin{aligned} \mathbf{A} &= \boldsymbol{\Psi}^{-1}\hat{\mathbf{A}}\boldsymbol{\Psi} = \begin{bmatrix} -10.0 & 10.0 & 4.334e-13 \\ 0.5 & -1.0 & 5.219e-13 \\ -5.254e-11 & 5.448e-11 & -2.667 \end{bmatrix}, \\ \mathbf{B} &= \boldsymbol{\Psi}^{-1}\hat{\mathbf{B}} = \begin{bmatrix} 1.0 \\ -2.207e-11 \\ 1.0 \end{bmatrix},\ \mathbf{C} = \hat{\mathbf{C}}\boldsymbol{\Psi} = \begin{bmatrix} 1.0 & -1.806e-11 & 1.0 \end{bmatrix}, \\ \mathbf{Q} &= \boldsymbol{\Psi}^{-1}\hat{\mathbf{Q}}(\boldsymbol{\Psi}\otimes\boldsymbol{\Psi}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0 & 0 & 0 & -0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \pm \boldsymbol{\epsilon}\cdot\mathbf{1^{3\times 9}}, \end{aligned} \tag{5.45}$$

where $\boldsymbol{\epsilon} \in [1e{-}12, 1e{-}10]$. The above result certifies that the original and identified systems are equivalent under a coordinate transformation (equivalent modulo). Important is also the fact that since $\mathbf{Q} \neq \boldsymbol{\Psi}^{-1}\hat{\mathbf{Q}}_s(\boldsymbol{\Psi}\otimes\boldsymbol{\Psi})$, quadratic identification with information from the first two kernels $H_1,\ H_2$ is impossible even if we have taken measurements with two harmonic input tones (off the diagonal). Here, the significant improvement compared with other similar efforts [104] is the systematic way of adding more information to the constructed model from the higher kernels. As a result, the forced Lorenz system was successfully identified when measurements of the first three symmetric kernels were considered, as illustrated in Fig. 5.3(left) in contrast with the unstable result obtained with information available only from the first two kernels. Finally, in Fig. 5.3(right), the identified and the original systems are equivalent state-space models after comparing them at the same coordinate system.

Case 2 - $\rho = 20$. For this case where $\rho > 1$, the Lorenz attractor has two non-zero equilibrium points from

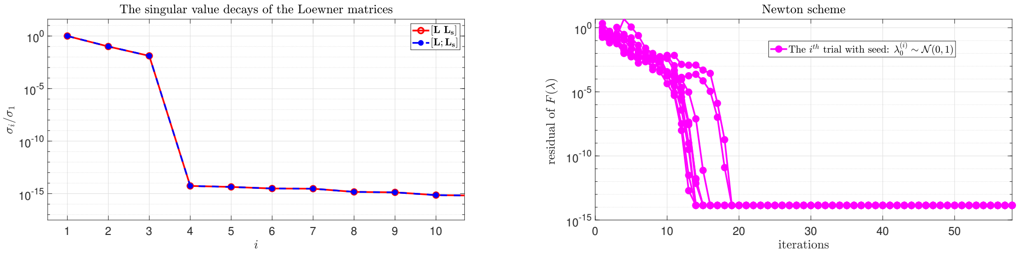


Figure 5.2.: Left: The Loewner singular value decay $r = 3$ with $\sigma_4/\sigma_1 \sim 1e - 14$. Right: The Newton convergence scheme. A solution vector $\boldsymbol{\lambda}^*$ has been obtained uniquely after starting with different random seeds $\boldsymbol{\lambda}_0$.

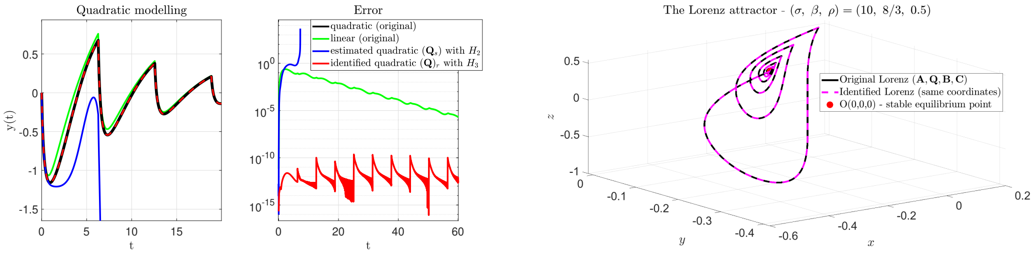


Figure 5.3.: Left: The linear model gives a poor approximation. Also, the H_2 does not contribute to a reasonable estimation of the quadratic operator; therefore, numerical instability is observed. After enhancing the information from the 3rd kernel, identification of the Lorenz system was achieved with a numerical error near machine precision. Right: The 3D state space is reconstructed from the identified system with the proposed method compared with the original one after aligning both systems to the same coordinates.

$$\begin{cases} 0 = -\sigma x(t) + \sigma y(t), \\ 0 = \rho x(t) - y(t) - x(t)z(t), \\ 0 = -\beta z(t) + x(t)y(t), \end{cases} \Rightarrow \mathbf{x}_e^{(1)} = \begin{bmatrix} \sqrt{\beta(\rho-1)} \\ \sqrt{\beta(\rho-1)} \\ \rho - 1 \end{bmatrix},\ \mathbf{x}_e^{(2)} = \begin{bmatrix} -\sqrt{\beta(\rho-1)} \\ -\sqrt{\beta(\rho-1)} \\ \rho - 1 \end{bmatrix} \tag{5.46}$$

Under harmonic excitation or non-zero initial conditions, the system's trajectories move around these two attractors. The chaotic behavior can be detected because the system can switch to a different steady state for small perturbations to initial conditions or the input, making the output evolution different.

Data assimilation over multiple steady-states. In Table 5.2, we show for a control system the way that measurements of the higher kernels can be obtained after exciting with harmonic inputs. Here, we illustrate this phenomenon by exciting with harmonic inputs the Lorenz attractor with the parameter $\rho = 20$. With $\alpha = 1$, $\omega_1 = 1$, we have two different designed complex inputs[4] that converge to the same input signal for large

[4] With complex inputs, e.g., $u(t) : \mathbb{R}_+ \to \mathbb{C}$, indexing harmonics and estimating kernels are straightforward tasks compared to the real input case, e.g., $u(t) : \mathbb{R}_+ \to \mathbb{R}$, where additional operations s.a.

t.

1. $\rho = 20$ - input 1: $u_1(t) = \underbrace{3e^{-0.1t}\texttt{sawtooth}(t)}_{\text{perturbation}} + \alpha e^{2j\pi\omega_1 t}$.

2. $\rho = 20$ - input 2: $u_2(t) = \alpha e^{2j\pi\omega_1 t}$.

As it is depicted in Fig. 5.4(left), for the different designed inputs $u_1(t)$, $u_2(t)$, we obtain two different steady-state solutions with different power spectrums Fig. 5.4(right). Measurements can be obtained for both systems, and the DC terms may help distinguish them. In Table 5.2, and for each system (blue, red), the Fourier spectrum (magnitude, phase) P provides the following measurements (complex inputs): $H_1(j\omega_1) = \frac{P_1(j\omega_1)}{a}$, $H_2(j\omega_1, j\omega_1) = \frac{P_2(j\omega_1,j\omega_1)}{a^2}$, $H_3(j\omega_1, j\omega_1, j\omega_1) = \frac{P_3(j\omega_1,j\omega_1,j\omega_1)}{a^3}$. For each system, the DC term, e.g., Eq. (5.38), can be computed from the non-periodic value $P(0)$ in the power spectrum. This can be generalized for multi-harmonic accurate signals (kernel separation) and in the X-parameter machinery [150] that deals with harmonic distortion. The symmetric transfer function that can interpret the measurements in Table 5.2 has to do with the corresponding linear operator, e.g., $\tilde{\mathbf{A}}_q = \mathbf{A} + 2\mathbf{Q}(\mathbf{x}_e^{(q)} \otimes \mathbf{I})$, $q = 1, 2$ and for each equilibrium respectively. For instance, using the equilibrium $\mathbf{x}_e^{(1)}$, we compute $\tilde{\mathbf{A}}_1 = \mathbf{A} + 2\mathbf{Q}(\mathbf{x}_e^{(1)} \otimes \mathbf{I})$.The 1st transfer function H_1 (for the equilibrium $\mathbf{x}_e^{(1)}$) yields the following value at frequency $\omega_1 = 2\pi$: $H_1(j\omega_1) = \mathbf{C}(j\omega_1\mathbf{I} - \tilde{\mathbf{A}})^{-1}\mathbf{B} = -0.0148 + 0.297\mathrm{i}$ which explains the measurements in the 1st row of Table 5.2, and similarly, the higher kernels explain the rest. Similar results can be obtained for the 2nd input-u_2 and the higher kernels. One way to distinguish different operational points (steady-states) among different equilibrium points is through the non-periodic term. For instance, when the

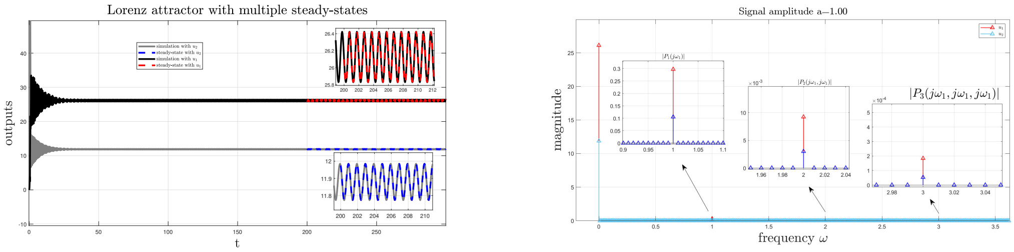


Figure 5.4.: Multiple steady-states and corresponding power spectrum. More details in the Table 5.2.

Data	DC	$H_1(j\omega_1)$	$H_2(j\omega_1, j\omega_2)$	$H_3(j\omega_1, j\omega_2, j\omega_3)$
$u_1(t)$	11.8819	$-0.0148 + 0.297\mathrm{i}$	$-0.00687 - 0.00614\mathrm{i}$	$1.74e - 4 - 5.82e - 5\mathrm{i}$
$u_2(t)$	26.1181	$0.09303 + 0.05011\mathrm{i}$	$-3.0e - 4 - 3.0e - 3\mathrm{i}$	$6.0e - 6 + 5.3e - 5\mathrm{i}$

Table 5.2.: Frequency spectrums under bifurcation that reveal multiple steady-states.

kernel separation with amplitude shifting, should be addressed [104, 37, 150].

Lorenz system is endowed with $\rho = 20 > 1$, we measure two different local quadratic systems that can be recognized from the two different DC terms in Fig. 5.4. Thus, the two different quadratic systems Eq. (5.21) can be identified, and the dynamics of the local coordinate system can be explained, with the respective equilibrium point as the origin. To discover the original model that has been bifurcated, we need to align the invariant operators to the same coordinates. Starting with a random seed for the Newton method, e.g., $\mathbf{T}_0 \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$, and applying Algorithm 5.3, we have the following convergence in Fig. 5.5 to the solution:

$$\mathbf{T}^{-1} = \begin{bmatrix} -0.003881 & -1.308 & -4.584 \\ -0.2823 & -0.8908 & -1.193 \\ -0.2035 & 0.6238 & 1.575 \end{bmatrix} \tag{5.47}$$

Now, the two quadratic systems have been aligned with the transformation $\mathbf{T}$, and the equilibrium points can be computed by solving Eq. (5.37) coupled with the information from the DC-terms Eq. (5.38) together with the enforcement of the operators $\hat{\mathbf{L}}$, $\check{\mathbf{L}}$ at the equilibrium points to be zero as analyzed in Eq. (5.40). Finally, by solving the above-coupled systems, we have the following results:

$$\boldsymbol{\lambda} = \begin{bmatrix} +20.45 \\ -58.29 \end{bmatrix}, \; \hat{\mathbf{x}}_e^{(1)} = \begin{bmatrix} 61.15 \\ 26.78 \\ 11.24 \end{bmatrix}, \; \hat{\mathbf{x}}_e^{(2)} = \begin{bmatrix} -16.28 \\ -45.49 \\ 11.39 \end{bmatrix}. \tag{5.48}$$

Having found the equilibrium points, we can derive the original linear operator from

$$\mathbf{A}_1 = \hat{\mathbf{A}}_1 - 2\hat{\mathbf{Q}}_1 \left(\mathbf{x}_e^{(1)} \otimes \mathbf{I} \right) = \begin{bmatrix} -7.873 & 7.255 & 66.07 \\ 3.056 & -7.245 & -48.22 \\ 1.602 & -1.405 & 1.451 \end{bmatrix}. \tag{5.49}$$

This linear operator $\mathbf{A}_1$ has the same eigenvalues as the original linear operator from the Lorenz system $\rho = 20$, $\texttt{eig}(\mathbf{A}_1) = \texttt{eig}(\mathbf{A}) = \begin{pmatrix} -20.34 & -2.667 & 9.341 \end{pmatrix}$. We could identify the original Lorenz system with the unstable linear operator with the proposed method. After transforming the operators $(\mathbf{A}_1, \mathbf{Q}_1, \mathbf{B}_1, \mathbf{C}_1)$ to the original coordinates, the two systems; original Eq. (5.43) and identified are exactly the same Fig. 5.5(right).

Case 3 - $\rho = 28$. Since for this parameter range (when $\rho > 24.74$), the dynamics are chaotic with the state evolution bifurcating from one equilibrium to the other without requiring additional energy, a steady state cannot be achieved, and measurements cannot be obtained for the higher kernels. Identifying such systems is not within the scope of this study. Generally, for any identification method, the operators can be identified to a finite numerical precision (e.g., IEEE machine precision $\epsilon \approx 2.22e - 16$). Since this is already an approximation with a non-zero numerical error, this slight numerical discrepancy will not allow any accurate prediction of such a sensitive system as the (deterministic) chaotic Lorenz attractor.

5.4.2. Reduction of the viscous Burgers' equation

In this example, we want to illustrate the proposed method in a larger-scale example. The aim is to construct robust surrogate models of a reduced order directly from physical measurements (i.e., samples of the symmetric GFRFs obtained from time-domain

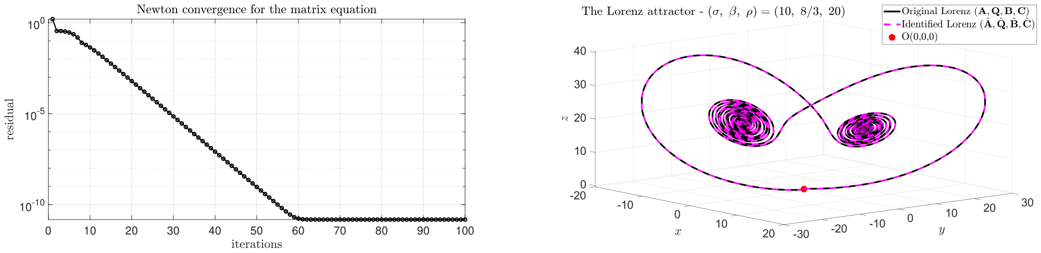


Figure 5.5.: Left: Convergence of the Newton scheme in Algorithm 5.3. Right: The original Lorenz system is identified with the two nontrivial equilibrium points. Here is the comparison between the original system and the identified one. The constructed state space evolution for both systems and at the same coordinates remains the same with zero numerical error.

simulations) that provide efficient approximations. A detailed description of the model under consideration can be found in [9]. We keep the same model set-up with a different viscosity parameter ν and observation space. Here, we consider as an output the velocity of the last tip of the flow $y(t) = x_{n+1}(t)$. Thus, the vector $\mathbf{C}$ contains everywhere zeros except from the last entry, which is 1. As illustrated in the study [9], the Loewner models for small viscosity coefficients ν may produce unstable results. As the current study relies on the Volterra series representation, analysis of the convergence with arbitrary viscosity and input amplitude remains an open issue. Hence, we illustrate a more conservative case with higher viscosity in what follows.

We use the problem data $\nu = 0.5$, $\sigma_0 = 0$, $\sigma_1 = 0.1$ representing the same physical quantities as in [9]. The full order model (FOM) is the linear finite element semi-discretization with $n = 257$. The semi-discretized system can result in Eq. (5.1) after inverting with the well-conditioned mass matrix $\mathbf{E}$. The system Eq. (5.1) solution is approximated with the Runge-Kutta multi-step integration method with a uniform time-discretization step $dt = 1/1000$. In the simulation bellow, we use $u_0(t) = 0.1e^{-0.2t}\texttt{sawtooth}(t) + 0.1\sin(4\pi t)$, and $u_1 \equiv 0$. Similarly, as in the Lorenz example, we take the following measurements:

- 100 logarithmic distributed measurements from the interval $[10^{-3}, 10^{1}]$,
- 400 logarithmic distributed measurements from the square grid $[10^{-3}, 10^{1}]^2$,
- 216 logarithmic distributed measurements from the cubic grid $[10^{-3}, 10^{1}]^3$.

In Fig. 5.7 (left), the singular value decay of the Loewner framework is presented. Thus, we choose the minimal linear order $r = 6$ with the first normalized truncated singular value of magnitude $\sigma_7/\sigma_1 = 5.10418 \cdot 1e - 10$. The recovery of the 1st GFRFs H_1 that results from the FOM with dimension $n = 257$ is compared with the reduced $\hat{H}_1$ of dimension $r = 6$ in Fig. 5.7(left).

Towards the estimation of the quadratic operator from the measurements of the 2nd GFRF H_2 (FOM) and after solving Eq. (5.9) with a threshold $\eta = 1e - 8$, we get the quadratic operator $\mathbf{Q}_s \in \mathbb{R}^{6\times 6^2}$. The hyper-parameter is tested for balancing the error with the norm of $\|\mathbf{Q}\|$ in a classical regularization sense. There are different ways to find the optimal regularization parameter η, e.g., Tikhonov regularization, L-curve that work similarly to the thresholding SVD. Moreover, the choice of η affects the length of the null space. In particular, out of $r^3 = 6^3 = 216$ degrees of freedom (DoF) and after enforcing the symmetries of the quadratic operator, the maximum rank could be $\texttt{rank} = 141 < 216$ when η is close to machine precision. Therefore, inverting with threshold $\eta = 1e-8$, the rank is 128, and the resulting null space has length $216 - 128 = 88$. These extra 88 free parameters will also be estimated to interpolate the 3rd GFRF. The fit performance between the 2nd level GFRFs from both FOM and ROM systems is illustrated in Fig. 5.6(left).

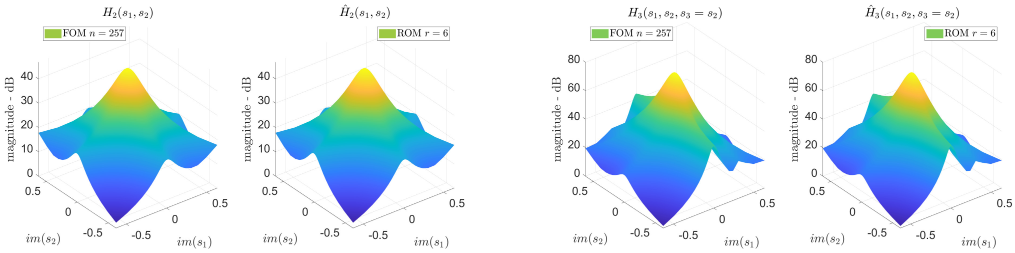


Figure 5.6.: Comparison between the FOM and ROM on the left, for the 2nd level kernels $\|\mathbf{H}_2(\mathbf{s}_1,\mathbf{s}_2) - \hat{\mathbf{H}}_2(\mathbf{s}_1,\mathbf{s}_2)\| = 3.1577e-05$, and on the right, for the 3rd level kernels. The error over the 2D plain-domain after fixing the 3rd dimension as $s_2 = s_3$ is $\|\mathbf{H}_3(\mathbf{s}_1,\mathbf{s}_2,\mathbf{s}_2 = \mathbf{s}_3) - \hat{\mathbf{H}}_3(\mathbf{s}_1,\mathbf{s}_2,\mathbf{s}_2 = \mathbf{s}_3)\| = 5.0353e-05$.

At this level, we estimate the quadratic operator denoted $\mathbf{Q}_s$ with information from the first two kernels H_1, H_2. Using the parameterization with λ_i, $i = 1,\ldots,88$ from Eq. (5.10), we enforce interpolation with the 3rd GFRF H_3 to estimate the remaining $m = 88$ parameters. Forming the data matrices in Algorithm 5.1, with hyper-parameter tuning as $\eta = 1e-9$, $\gamma_0 = 1e-5$, the residue of the Newton iterations stagnates to the value $2.2478e-06$. In Fig. 5.6(right) comparison between the 3rd level kernels FOM($n = 257$) and ROM($r = 6$) is depicted.

Finally, in Fig. 5.7, both systems FOM($n = 257$) and ROM($r = 6$) are compared under a nontrivial input and for an extended simulation that covers all the dynamic evolution starting from a hard transient up to the steady-state profile. By considering more measurements from higher-order kernels, the fitting performance improves significantly (for the complete time interval of the simulation). The proposed method achieves approximate interpolation to all the measurement data sets, and the accuracy performance is improved when using the three kernels Fig. 5.7(right). The updated quadratic operator $\hat{\mathbf{Q}}$ of dimension 6×6^2 interpolates approximately the 3rd kernel. To illustrate this result, in Table 5.3, we choose a random point in the three-dimensional frequency space, and we test the interpolation error for both estimations of the quadratic operator; firstly, from the two kernels as $\mathbf{Q}_s$; secondly, from the three kernels as $\mathbf{Q}_r$.

In this study, we were concerned with identifying or constructing quadratic state-space

Kernels & frequencies	Evaluation at $(s_1, s_2, s_3) = (1\mathrm{i}, 2\mathrm{i}, 3\mathrm{i})$	Interpolation with the FOM
FOM $H_2(s_1, s_2)$	$-0.20829 + 0.13846\mathrm{i}$	theoretical value
$\hat{H}_2(s_1, s_2, \mathbf{Q}_r)$	$-0.20829 + 0.13846\mathrm{i}$	✓
$\hat{H}_2(s_1, s_2, \mathbf{Q}_s)$	$-0.20829 + 0.13846\mathrm{i}$	✓
FOM $H_3(s_1, s_2, s_3)$	$0.042016 + 0.027069\mathrm{i}$	theoretical value
$\hat{H}_3(s_1, s_2, s_3, \mathbf{Q}_r, \mathbf{Q}_r)$	$0.042015 + 0.027069\mathrm{i}$	✓
$\hat{H}_3(s_1, s_2, s_3, \mathbf{Q}_s, \mathbf{Q}_s)$	$0.031301 + 0.00015172\mathrm{i}$	×

Table 5.3.: Symmetric Volterra kernel interpolation at a random point. The updated $\mathbf{Q}_r$ from the three kernels enforces interpolation to the 3rd kernel without ruining the interpolation on the 2nd kernel. As a result, the overall performance has improved significantly Fig. 5.7(right).

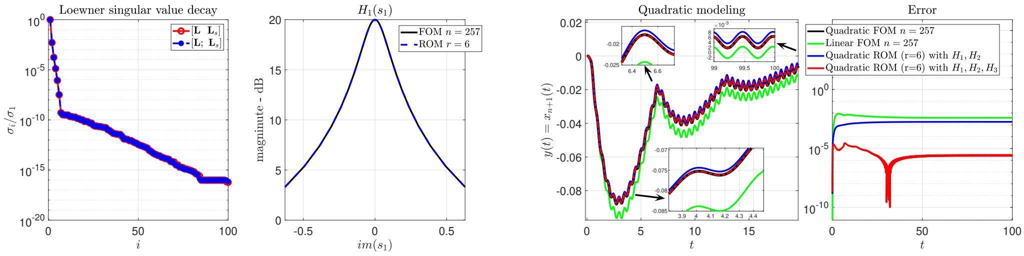


Figure 5.7.: From left to right, the singular value decay of the Loewner matrix. Comparison between the FOM and ROM 1st level kernels with error $\|\mathbf{H}_1(\mathbf{s}_1) - \hat{\mathbf{H}}_1(\mathbf{s}_1)\| = 8.9852e - 09$. Reduction of a percentage level 98% has been achieved with the error between the FOM($n = 257$) and ROM($r = 6$) to be $\|\hat{y}(t) - y(t)\| \approx 1e - 5$.

models from i/o time domain data. Such models can be obtained from the first principles, e.g., Newtonian dynamics that result in second-order $\ddot{\mathbf{x}}(t) \in \mathbb{R}^n$, and after transforming equivalently to first-order $\dot{\mathbf{x}}(t) \in \mathbb{R}^{2n}$, we result in systems with ODEs of a specific nonlinear degree. For instance, dynamical systems that belong to the class of quadratic control are; Navier Stokes, Burgers' equation, Lorenz attractor, etc. Using the symmetric generalized frequency Volterra kernels that can be estimated from a physical system under input-output harmonic excitation, the proposed method identifies/constructs quadratic models. By having estimations of a finite set out of the infinite Volterra kernels, and after enforcing interpolation, e.g., to the first three (H_1, H_2, H_3), the resulting quadratic system inherits a Volterra series that interpolates the original one to a specific set of chosen frequency points for the first kernels and approximates the rest of the infinite terms that eventually decay to negligible dynamics. The proposed method is not limited to systems bifurcating into different equilibrium points. The steady-state measurements explain the local behavior of the phenomenon without ensuring that the actual dynamics are not described from a global model that bifurcates to different equilibrium points. Therefore, we have illustrated this phenomenon from the forced Lorenz attractor model with parameters that produce this effect. With the proposed method, we identified the global model of the Lorenz attractor from i/o time domain data and with parameters

($\sigma = 10,\ \beta = 8/3,\ \rho = 20$), after taking care of the invariant information that carries along the different equilibrium points. The proposed method has been tested w.r.t. the reduction performance for a larger-scale example (the Burgers' equation), and a quadratic surrogate model has been constructed of order $r = 6$ that achieved 98% reduction and accuracy close to 5 digits.

For systems that involve different nonlinear dynamics, such as classical oscillators, e.g., Duffing, Van der Pol, etc., the same analysis can be derived for polynomial state-space systems to a specific nonlinear degree s.a. cubic order, i.e., $\mathbf{x}\otimes\mathbf{x}\otimes\mathbf{x}$ that physically can explain nonlinear stiffness and damping. Lifting strategies for equivalently representing nonlinear systems with analytical nonlinearities to the quadratic form are left for future research endeavors. The main difficulty for non-intrusive methods like the one presented here is dealing with a "partially missing" linear operator $\mathbf{A}$. Therefore, we aim to analyze such phenomena in the future, i.e., for which the resolvent $\mathbf{\Phi}(s) = (s\mathbf{I} - \mathbf{A})^{-1}$ contains a sparse linear operator $\mathbf{A}$, that may contain many zero diagonal blocks (due to, e.g., applying lifting approaches). Although the tools used in this study are robust to noise (such as most spectral transforms), more involved analysis of the impact of the noise is left for future studies. Moreover, we plan to involve machine learning techniques that can be advantageous to methods such as the proposed one due to their power to learn nonlinear i/o maps (universal approximation theorem). For instance, when solely one input-output sequence of measurements is accessible (and not many such sequences), a neural network (NN) can be used as a surrogate black box model for transferring the whole measurement process to more efficient, cheap simulations. Finally, connecting data and computational science tools, e.g., NNs, with the proposed method will increase interpretability for ML tools. More precisely, by constructing interpretable state-space dynamic models, for which the analysis has matured over many decades, ad-hoc engineering practices will become more reliable.

5.5. Quadratic-bilinear modeling from i/o time-domain data

We analyze in what follows dynamical systems as Eq. (2.45) with $\mathbf{f}(\mathbf{x}(t)) = \mathbf{Q}(\mathbf{x}(t)\otimes\mathbf{x}(t))$ and $\mathbf{g}(\mathbf{x}(t)) = \mathbf{N}\mathbf{x}(t) + \mathbf{B}$.

5.5.1. Quadratic-bilinear control systems and properties

Let the state-space representation of such a system be given as:

$$\begin{cases}\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{Q}(\mathbf{x}(t)\otimes\mathbf{x}(t)) + \mathbf{N}\mathbf{x}(t)u(t) + \mathbf{B}u(t),\\ \quad y(t) = \mathbf{C}\mathbf{x}(t),\end{cases}, \tag{5.50}$$

where $\mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0}$ and the matrix $\mathbf{E} \in \mathbb{R}^{n\times n}$ is non-singular, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{Q} \in \mathbb{R}^{n\times n^2}$, $\mathbf{N} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times 1}$ and $\mathbf{C} \in \mathbb{R}^{1\times n}$. Moreover, assume that $\mathbf{Q}$ satisfies the property $\mathbf{Q}(\mathbf{v}\otimes\mathbf{w}) = \mathbf{Q}(\mathbf{w}\otimes\mathbf{v})$, i.e., it is represented in a "symmetrizable format."

The first two generalized symmetric transfer functions of a QB system as in (5.50) are:

$$\begin{aligned} H_1(s_1) &= \mathbf{C}_\ell \boldsymbol{\Phi}(s_1)\mathbf{B}, \\ H_2(s_1,s_2) &= \mathbf{C}_\ell \boldsymbol{\Phi}(s_1+s_2)\mathbf{Q}(\boldsymbol{\Phi}(s_1)\mathbf{B}\otimes\boldsymbol{\Phi}(s_2)\mathbf{B}) \\ &\quad + \frac{1}{2}\mathbf{C}\boldsymbol{\Phi}(s_1+s_2)\mathbf{N}(\boldsymbol{\Phi}(s_1)\mathbf{B}+\boldsymbol{\Phi}(s_2)\mathbf{B}). \end{aligned} \tag{5.51}$$

For more details on deriving such functions, we refer the reader to [38] and to [68, 9], which work similarly as we have illustrated for the bilinear and the quadratic cases separately. Once more, let us emphasize that the essential property of these particular functions (symmetric transfer functions) is that their samples can be inferred from the spectrum of the observed output when using a purely oscillatory control input.

5.5.2. Fitting quadratic-bilinear models from i/o harmonic data

The idea is to recover all the operators in Eq. (5.50) from physical time domain measurements. The Loewner framework can recover the linear part by providing a fitted realization of dimension r, denoted with $(\hat{\mathbf{E}},\hat{\mathbf{A}},\hat{\mathbf{B}},\hat{\mathbf{C}})$. Now, based on these matrices, introduce $\hat{\boldsymbol{\Phi}}(s) = (s\hat{\mathbf{E}}-\hat{\mathbf{A}})^{-1}\in\mathbb{C}^{r\times r}$.

The sampling domain is denoted with $\Omega = \{(\zeta_1^{(i)},\zeta_2^{(i)})\in\mathbb{C}^2 | 1\leq i\leq K\}$. For a particular pair of sampling points $(\zeta_1^{(i)},\zeta_2^{(i)})\in\mathbb{C}^2$, we define the following quantities:

$$\begin{aligned} \hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)}) &:= \hat{\mathbf{C}}\hat{\boldsymbol{\Phi}}(\zeta_1^{(i)}+\zeta_2^{(i)})\in\mathbb{C}^{1\times r}, \\ \hat{\mathcal{R}}_q(\zeta_1^{(i)},\zeta_2^{(i)}) &:= (\hat{\boldsymbol{\Phi}}(\zeta_1^{(i)})\hat{\mathbf{B}}\otimes\hat{\boldsymbol{\Phi}}(\zeta_2^{(i)})\hat{\mathbf{B}})\in\mathbb{C}^{r^2\times 1}, \\ \hat{\mathcal{R}}_b(\zeta_1^{(i)},\zeta_2^{(i)}) &:= (\hat{\boldsymbol{\Phi}}(\zeta_1^{(i)})\hat{\mathbf{B}}+\hat{\boldsymbol{\Phi}}(\zeta_2^{(i)})\mathbf{B})\in\mathbb{C}^{r\times 1}. \end{aligned} \tag{5.52}$$

It is to be noted that the vectors introduced in (5.52) are computed solely in terms of the matrices $(\hat{\mathbf{E}},\hat{\mathbf{A}},\hat{\mathbf{B}},\hat{\mathbf{C}})$ corresponding to the data-driven Loewner surrogate model. Let $\mathbf{v}\in\mathbb{C}^K$ be the vector of data measurements, i.e., containing samples of the second symmetric transfer function $H_2(s_1,s_2)$ evaluated on the Ω grid. More precisely, let $\mathbf{v}_i = H_2(\zeta_1^{(i)},\zeta_2^{(i)})$. Now, since we would like to fit a reduced-order QB model to interpolate the 2D data, it follows that $H_2(\zeta_1^{(i)},\zeta_2^{(i)}) = \hat{H}_2(\zeta_1^{(i)},\zeta_2^{(i)})$. We can write:

$$\underbrace{\hat{H}_2(\zeta_1^{(i)},\zeta_2^{(i)})}_{\mathbf{v}_i\in\mathbb{C}} = \underbrace{\hat{\mathbf{C}}\hat{\boldsymbol{\Phi}}(s_1+s_2)}_{\hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)})\in\mathbb{C}^{1\times r}}\ \hat{\mathbf{Q}}\ \underbrace{(\hat{\boldsymbol{\Phi}}(\zeta_1^{(i)})\hat{\mathbf{B}}\otimes\hat{\boldsymbol{\Phi}}(s_2)\hat{\mathbf{B}})}_{\hat{\mathcal{R}}_q(\zeta_1^{(i)},\zeta_2^{(i)})\in\mathbb{C}^{r^2\times 1}} \tag{5.53}$$

$$+\,(1/2)\,\underbrace{\mathbf{C}\boldsymbol{\Phi}(\zeta_1^{(i)}+\zeta_2^{(i)})}_{\hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)})\in\mathbb{C}^{1\times r}}\hat{\mathbf{N}}\,\underbrace{(\hat{\boldsymbol{\Phi}}(\zeta_1^{(i)})\hat{\mathbf{B}}+\hat{\boldsymbol{\Phi}}(\zeta_2^{(i)})\mathbf{B})}_{\hat{\mathcal{R}}_b(\zeta_1^{(i)},\zeta_2^{(i)})\in\mathbb{C}^{r\times 1}}, \tag{5.54}$$

and hence it follows that:

$$\hat{\mathbf{H}}_2(\zeta_1^{(i)},\zeta_2^{(i)}) = \hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)})\hat{\mathbf{Q}}\hat{\mathcal{R}}_q(\zeta_1^{(i)},\zeta_2^{(i)}) \tag{5.55}$$

$$+\,(1/2)\hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)})\hat{\mathbf{N}}\hat{\mathcal{R}}_b(\zeta_1^{(i)},\zeta_2^{(i)}) \tag{5.56}$$

Definition 5.7 (Vectorization operator):
Given a matrix $\mathbf{X} \in \mathbb{C}^{m\times n}$, we denote with $\text{vec}(\mathbf{X})$ the vector $(mn) \times 1$ computed as follows:

$$\text{vec}(\mathbf{X}) = \begin{bmatrix} \mathbf{X}(1,:) & \cdots & \mathbf{X}(m,:) \end{bmatrix}^T \in \mathbb{C}^{mn}, \tag{5.57}$$

where the MATLAB notation $\mathbf{X}(k,:) \in \mathbb{C}^{1\times n}$ was used to refer to the kth row of $\mathbf{X}$. $\Diamond$

The vectorization procedure adapted to the data-driven problem is presented in (5.59). Let us denote with $\mathcal{T} \in \mathbb{C}^{K\times(r^3+r^2)}$, the matrix for which the ith row is given by $\mathcal{T}(i,:) = \begin{bmatrix}\mathcal{T}_q(i,:) & \mathcal{T}_b(i,:)\end{bmatrix} \in \mathbb{C}^{1\times(r^3+r^2)}$

$$\begin{cases} \mathcal{T}_q(i,:) = \begin{bmatrix} \hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)}) \otimes \hat{\mathcal{R}}_q^T(\zeta_1^{(i)},\zeta_2^{(i)}) \end{bmatrix} \in \mathbb{C}^{K\times r^3}, \\ \mathcal{T}_b(i,:) = \begin{bmatrix} \hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)}) \otimes \hat{\mathcal{R}}_b^T(\zeta_1^{(i)},\zeta_2^{(i)}) \end{bmatrix} \in \mathbb{C}^{K\times r^2}. \end{cases} \tag{5.58}$$

$$\begin{aligned} \mathbf{v}_i &= \hat{H}_2(\zeta_1^{(i)},\zeta_2^{(i)}) = \hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)})\hat{\mathbf{Q}}\hat{\mathcal{R}}_q(\zeta_1^{(i)},\zeta_2^{(i)}) + (1/2)\hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)})\hat{\mathbf{N}}\hat{\mathcal{R}}_b(\zeta_1^{(i)},\zeta_2^{(i)}) \\ &= \left(\hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)}) \otimes \hat{\mathcal{R}}_q^T(\zeta_1^{(i)},\zeta_2^{(i)})\right) vec(\hat{\mathbf{Q}}) + (1/2)\left(\hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)}) \otimes \hat{\mathcal{R}}_b^T(\zeta_1^{(i)},\zeta_2^{(i)})\right) vec(\hat{\mathbf{N}}) \\ &= \underbrace{\left[\begin{array}{c|c} \underbrace{\hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)}) \otimes \hat{\mathcal{R}}_q^T(\zeta_1^{(i)},\zeta_2^{(i)})}_{=\mathcal{T}_q(i,:)} & \underbrace{\hat{\mathcal{O}}(\zeta_1^{(i)},\zeta_2^{(i)}) \otimes \hat{\mathcal{R}}_b^T(\zeta_1^{(i)},\zeta_2^{(i)})}_{=\mathcal{T}_b(i,:)} \end{array}\right]}_{=\mathcal{T}(i,:)\ \in\ \mathbb{C}^{1\times(r^3+r^2)}} \cdot \underbrace{\left[\begin{array}{c} \text{vec}(\hat{\mathbf{Q}}) \\ \hline (1/2)\text{vec}(\hat{\mathbf{N}}) \end{array}\right]}_{=\mathbf{z}\ \in\ \mathbb{C}^{(r^3+r^2)\times 1}} \end{aligned} \tag{5.59}$$

Now, from (5.59), by varying the index i such as $1 \le i \le K$, it follows that we can put together a linear system of equations in $r^3 + r^2$ unknowns as follows:

$$\mathcal{T}\mathbf{z} = \mathbf{v}, \tag{5.60}$$

where $\mathbf{z} = \begin{bmatrix}\text{vec}(\hat{\mathbf{Q}})^T & (1/2)\text{vec}(\hat{\mathbf{N}})^T\end{bmatrix}^T$ is the vector of variables that contains the entries of the vectorized operators of the surrogate reduced-order QB system. We need enough data measurements to ensure an over-determined linear system of equations, i.e., the condition $K \ge r^3 + r^2$ needs to hold. Then, we can employ a direct solution of system (5.60), e.g., using the Moore-Penrose pseudo-inverse or by using Gaussian elimination. If the matrix $\mathcal{T}$ has full column rank, then it does not matter what procedure is chosen (for the direct solve). However, in most cases, the matrix $\mathcal{T}$ is not of full column rank; hence, direct solutions must be carefully dealt with (by introducing regularization techniques). In what follows, we will use a truncated singular value decomposition (tSVD) approach. This is an attractive and powerful method since it uses the optimal rank-k approximation of the SVD (in the 2 norms). Such an approach has already been used for applying OpInf [27], together with the Tikhonov regularization scheme [128] and the tQR (truncated QR decomposition) approach.

Remark 5.8 (Regularization):
Finding a viable solution for the least-squares problem stated in (5.60) is not a straightforward task and can be a computationally challenging task because of the ill-conditioning of matrix $\mathcal{T}$ (the column rank of matrix $\mathcal{T}$ is sometimes much smaller than $r^3 + r^2$). Another challenge is the computational cost which grows cubically in the order r of the reduced-order system. $\Diamond$

5.5.3. A nonlinear RC Ladder circuit

We analyze a nonlinear RC-ladder electronic circuit first introduced in [46]. Various variants of this model were also mentioned in other MOR works,i.e., [86] and [38]. This nonlinear first-order system models a resistor-capacitor network that exhibits a nonlinear behavior caused by the nonlinear resistors consisting of a parallel connected resistor with a diode. As presented in [46], a SISO system of the form gives the underlying model:

$$\dot{x}(t) = \begin{bmatrix} -g(x_1(t)) - g(x_1(t) - x_2(t)) \\ g(x_1(t) - x_2(t)) - g(x_2(t) - x_3(t)) \\ \vdots \\ g(x_{k-1}(t) - x_k(t)) - g(x_k(t) - x_{x+1}(t)) \\ \vdots \\ g(x_{N-1}(t) - x_N(t)) \end{bmatrix} + \begin{bmatrix} u(t) \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{5.61}$$

with $y(t) = x_1(t)$, where the mapping g is given by $g : \mathbb{R} \to \mathbb{R}$ defined as $g(x_i) = g_D(x_i) + x_i$, which combines the effect of a diode and a resistor. The non-linearity g_D models a diode as a nonlinear resistor, based on the classical Shockley model:

$$g_D(x_i) = i_S(\exp(u_P x_i) - 1), \tag{5.62}$$

with material parameters $i_S > 0$ and $u_P > 0$. For this benchmark, the parameters are selected as follows: $i_S = 1$ and $u_P = 40$ as in [46]. By substituting these values into (5.62), we get that $g_D(x_i) = \exp(40x_i) - 1$, and hence it follows that $g(x_i) = \exp(40x_i) + x_i - 1$.

In what follows, we will apply the proposed methods for:

5.5.3.1. Linear approximation

By eliminating the higher-order terms from the Taylor expansion, we are using here the following linear approximation to g_D, i.e., $g_D(x_i) = 40x_i$, and hence $g(x_i) = g_1(x_i) = 41x_i$.

$$\dot{x}(t) = \begin{bmatrix} -82x_1(t) + 41x_2(t) \\ 41x_1(t) - 82x_2(t) + 41x_3(t) \\ \vdots \\ 41x_{k-1}(t) - 82x_k(t) + 41x_{k+1}(t) \\ \vdots \\ 41x_{N-1}(t) - 41x_N(t) \end{bmatrix} + \begin{bmatrix} u(t) \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{5.63}$$

5.5.3.2. Quadratic approximation

By eliminating the higher-order terms starting with the cubic ones, from the Taylor expansion, we are using here the following linear approximation to g_D, i.e., $g_D(x_i) = 40x_i + 800x_i^2$, and hence

$$g(x_i) = g_1(x_i) = 41x_i + 800x_i^2. \tag{5.64}$$

5.5.3.3. Bilinear treatment via Carleman's approach

The original nonlinear system is transformed into a bilinear system employing Carleman linearization, as originally shown in [46] and later in [38]. The matrices are:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{A}_1 & \frac{1}{2}\mathbf{A}_2 \\ \mathbf{0} & \mathbf{A}_1 \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}_1 \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \mathbf{v} \otimes \mathbf{v} \end{bmatrix}, \\ \mathbf{N} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{B} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{B} & \mathbf{0} \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} \mathbf{c} \\ \mathbf{0} \end{bmatrix}^T. \end{aligned} \tag{5.65}$$

Consequently, the resulting bilinear system has dimension $n^2 + n$ with n as the number of circuit blocks of the original system. More details on the structure of the involved matrices in Eq. (5.65) can be found in [38].

5.5.3.4. Lifting to quadratic-bilinear form

The original RC-ladder model can be lifted to an equivalent quadratic bilinear model. The introduced additional state variables $\mathbf{x}_1 = v_1$ and $x_i = v_i - v_{i+1}$ followed by introducing the additional state variables $z_1 = e^{-40v_1} - 1$ and $z_i = e^{40x_i}$ can transform equivalently the original system Eq. (5.61) to a quadratic-bilinear form Eq. (5.50) with dimension $2n$ [38].

In Fig. 5.8, the original nonlinear system, the equivalent quadratic-bilinear, and the approximated bilinear are depicted. The numerical difference between the original and the quadratic-bilinear has reached machine precision, where the bilinear for this amplitude starts to differ significantly.

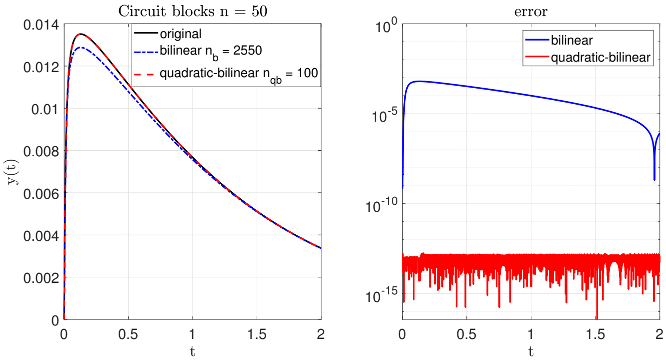


Figure 5.8.: The original RC-ladder model with $n = 50$ circuit blocks. The lifted QB model is equivalent to the original nonlinear. The bilinear model of dimension $n^2 + n$ offers good approximation only for relatively small input amplitudes.

The aim is to achieve MOR by measuring the first symmetric Volterra kernels from input-output time-domain simulations. A double-tone input is considered as we infer the operators from the 2nd kernel. The scheme for kernel separation and harmonic indexing remains the same as in [104]. Simulating the original nonlinear model in the time domain under the excitation of a double-tone harmonic input can make an accurate separation of kernels (the Fourier transform is indeed accurate). Here, as we want to illustrate the efficacy of the proposed method by inferring the operators from the 2nd Kernel, we assume a perfect measurement setup.

Step 1: The first harmonic can be measured under excitation with a single-tone input. Then, measurements of the first kernel $H_1(s_1)$ (e.g., the magnitude and phase) can be derived. The Loewner framework constructs a low-order rational interpolant and identifies the minimal linear sub-system of order r. In Fig. 5.3 and on the left pane, the Loewner singular value decay offers the criterion for reduction. The order $r = 10$ was chosen, as the 11th singular value is close to machine precision. Hence, it is an indicator for recovering the original linear dynamics. Denote with $\Sigma_{\texttt{lin}} : (\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}})$ the linear reduced system of order $r = 10$.

The approximation results are depicted in Fig. 5.9 and on the right pane.

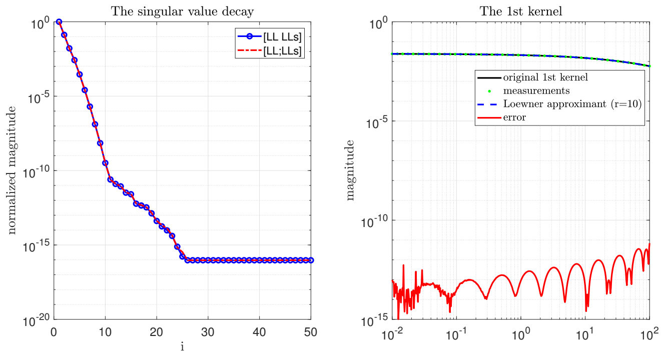


Figure 5.9.: Left pane: The Loewner singular value decay (used to decide on the reduced order). Right pane: The reduced-order Loewner interpolant computed from the 1st kernel can reach machine precision approximation.

Step 2: By having access to the reduced operators of the linear system $\hat{\Sigma}_{\texttt{lin}}$, we can infer the remaining nonlinear quadratic and bilinear operators from Eq. (5.59). The second harmonic can be measured with a double-tone excitation. Thus, measurements of the second kernel $H_2(s_1, s_2)$ over the whole complex domain of definition can be collected repetitively. It is important to mention that the amount of measurements is related to the reduced dimension r, where at least $K \geq (r^3 + r^2)$ measurements ensure enough data for the solution of Eq. (5.60). By enforcing quadratic symmetries (the matrix $\hat{\mathbf{Q}}$ is set to satisfy the property $\hat{\mathbf{Q}}(\mathbf{w} \otimes \mathbf{v}) = \hat{\mathbf{Q}}(\mathbf{v} \otimes \mathbf{w})$), the complexity can be further reduced. Another simplification can be performed using some algebraic adjustments by replacing the symmetric Kronecker product with the asymmetric one ($\otimes'$ as in [28]). Solving for the vector $\mathbf{z}$ yields a reduced-order QB system $\hat{\Sigma}_{QB} = (\hat{\mathbf{A}}, \hat{\mathbf{Q}}, \hat{\mathbf{N}}, \hat{\mathbf{B}}, \hat{\mathbf{C}})$. Finally, inFig. 5.10(right), the time domain solution is depicted and compared to the original nonlinear response.

The proposed method successfully constructs a reduced quadratic-bilinear model from input-output time domain data that accurately approximates the response of the original nonlinear system. The simulations are performed with the multi-step Runge Kutta

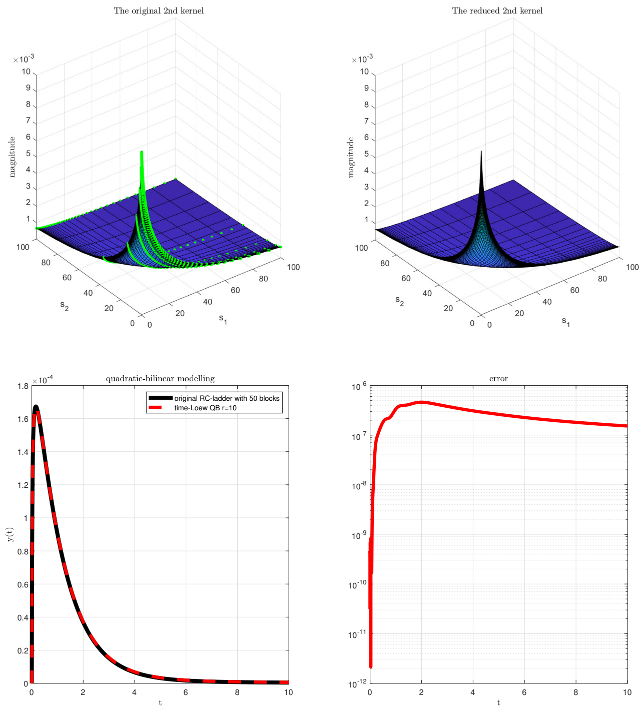


Figure 5.10.: Left: The original 2nd kernel and the measurements (green dots) compared with the reduced kernel $\|H_2(s_1, s_2) - \hat{H}_2(s_1, s_2)\|_\infty \sim 10^{-8}$. Right: The original nonlinear RC ladder and the approximant were constructed with the new LoewQB method. The absolute error over a dense grid.

scheme (e.g., using `ODE45` in Matlab) with input $u(t) = 0.01e^{-t}$, and the maximum error is $\|(y(t) - y_r(t))\|_\infty \sim 10^{-6}$.

A non-intrusive data-driven method that constructs nonlinear models with the quadratic-bilinear structure from input-output time-domain data was presented. The method was based on the Loewner and Volterra frameworks. The second symmetric kernel can be inferred as the 2nd harmonic of the transformed output; hence, the problem of estimating the nonlinear operators can be resolved in a linear LS system. Although this LS system may be under-determined, it contains enough information for estimating the nonlinear operators with the Moore-Penrose pseudo-inverse (truncated SVD), thus providing good approximations. The use of higher harmonics (e.g., 3rd kernel, etc.) that results in a nonlinear optimization problem, along with the use of other regularization techniques as the one proposed by Tikhonov, will be the topics of future research endeavors in connection also with the analysis on the sensitivity of noise that actual data contain.

Algorithm 5.4: Quadratic-bilinear modeling from input-output time data

```
Input: Measurements of the first two symmetric kernels $H_1(s_1),\ H_2(s_1,s_2)$
Output: A quadratic bilinear system $(\mathbf{A}_r,\mathbf{Q}_r,\mathbf{N}_r,\mathbf{B}_r,\mathbf{C}_r)$
1: Construct the linear minimal subsystem $(\mathbf{A}_r,\mathbf{B}_r,\mathbf{C}_r)$ with the Loewner framework
   Algorithm 3.1 from $H_1(s_1)$
2: Solve the system in Eq. (5.60) with measurements of $H_2(s_1,s_2)$ and get the
   quadratic and bilinear operators $\mathbf{Q}_r,\mathbf{N}_r$
```

5.6. Spectral quadratic-bilinear identification from state access

Consider the quadratic-bilinear system of order n that is described in the following ODE

$$\mathbf{\Sigma}_{QB}:\dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t)+\mathbf{Q}(\mathbf{x}(t)\otimes\mathbf{x}(t))+\mathbf{N}\mathbf{x}(t)u(t)+\mathbf{B}u(t),\ \mathbf{x}(0)=\mathbf{x}_0=\mathbf{0}. \tag{5.66}$$

We consider the general m-tone complex input as

$$u(t)=\sum_{i=1}^{m}e^{s_it}, \tag{5.67}$$

and with the growing exponential approach, the steady-state solution can be assumed as

$$\mathbf{x}_{ss}(t)=\sum_{k_1=0}^{\infty}\cdots\sum_{k_m=0}^{\infty}\mathbf{G}_{(k_1+k_2+\cdots+k_m)}(k_1s_1,k_2s_2,\ldots,k_ms_m)e^{(k_1s_1+k_2s_2+\cdots+k_ms_m)t}. \tag{5.68}$$

Remark 5.9 (The mth symmetric Volterra kernel):
To define the mth symmetric Volterra kernel, the system has to be excited with an mth-tone harmonic input. ♢

In the Appendix B, we analyze the derivation of the higher symmetric Volterra kernels for the quadratic-bilinear control systems.

The aim is the identification of the operators $(\mathbf{A},\mathbf{Q},\mathbf{N},\mathbf{B})$ from the symmetric $\mathbf{G}$s by having state access, so the order is a priori known as the dimension of the system n. For the QB system with dimension n, the total parameters to be determined are $(n^2+n^3+n^2+n)$, so in total, $K=n^3+2n^2+n$. The Hessian $\mathbf{Q}$ has the symmetric property $\mathbf{Q}(\mathbf{x}_1\otimes\mathbf{x}_2)=\mathbf{Q}(\mathbf{x}_2\otimes\mathbf{x}_1)$ and with that only $n^2(n+1)/2$ should be determined.

Measurements of the Gs can be obtained via time-domain simulation of the original system lifted to a QB.

We start with the first kernel,

$$\begin{aligned}
&\mathbf{G}_1(s_1) = (s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} \Rightarrow (s_1\mathbf{I}_n - \mathbf{A})\mathbf{G}_1(s_1) = \mathbf{B} \Rightarrow \\
&\mathbf{A}\mathbf{G}_1(s_1) + \mathbf{B} = s_1\mathbf{G}_1(s_1) \Rightarrow \mathbf{I}_n\mathbf{A}\mathbf{G}_1(s_1) + \mathbf{B} = s_1\mathbf{G}_1(s_1) \Rightarrow \\
&(\mathbf{I}_n \otimes \mathbf{G}_1^T(s_1))vec(\mathbf{A}) + \mathbf{B} = s_1\mathbf{G}_1(s_1) \Rightarrow \\
&\begin{bmatrix} \mathbf{I}_n \otimes \mathbf{G}_1^T(s_1) & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} vec(\mathbf{A}) \\ \mathbf{B} \end{bmatrix} = s_1\mathbf{G}_1(s_1)
\end{aligned} \tag{5.69}$$

A solution can be obtained from the above least squares problem with the pseudo-inverse when the linear controllability matrix is not full rank. The recovery of $\mathbf{B}$ is exact, and an estimation of the linear operator is denoted as $\hat{\mathbf{A}}$, while the 1st kernel is $\hat{\mathbf{G}}_1 = (s\mathbf{I}_n - \hat{\mathbf{A}})^{-1}\mathbf{B}$ with $rank([s\mathbf{I}_n - \hat{\mathbf{A}}]) = n$.

For the rest of the kernels, with $\mathbf{\Phi}(s_1 + \cdots + s_m) = ((s_1 + \cdots + s_m)\mathbf{I}_n - \mathbf{A})^{-1}$, we can write them arbitrarily with the $\mathbf{\Phi}, \mathbf{R_Q}, \mathbf{R_N}$ notation as:

$$\mathbf{G}_m(s_1, \ldots, s_m) = \mathbf{\Phi}(s_1 + \cdots + s_m)\mathbf{Q}\mathbf{R_Q}(s_1, \ldots, s_m) + \mathbf{\Phi}(s_1 + \cdots + s_m)\mathbf{N}\mathbf{R_N}(s_1, \ldots, s_m). \tag{5.70}$$

Dropping $\mathbf{\Phi}$ on the left-hand side and using Kronecker product properties yields:

$$\begin{aligned}
&\mathbf{\Phi}^{-1}(s_1 + \cdots + s_m)\mathbf{G}_m(s_1, \ldots, s_m) = \mathbf{Q}\mathbf{R_Q}(s_1, \ldots, s_m) + \mathbf{N}\mathbf{R_N}(s_1, \ldots, s_m) \Rightarrow \\
&\Big((s_1 + \cdots + s_m)\mathbf{I}_n - \mathbf{A}\Big)\mathbf{G}_m(s_1, \ldots, s_m) = \mathbf{I}_n\mathbf{Q}\mathbf{R_Q}(s_1, \ldots, s_m) + \mathbf{I}_n\mathbf{N}\mathbf{R_N}(s_1, \ldots, s_m) \Rightarrow \\
&(s_1 + \cdots + s_m)\mathbf{G}_m(s_1, \ldots, s_m) - \mathbf{I}_n\mathbf{A}\mathbf{G}_m(s_1, \ldots, s_m) = \mathbf{I}_n\mathbf{Q}\mathbf{R_Q}(s_1, \ldots, s_m) + \mathbf{I}_n\mathbf{N}\mathbf{R_N}(s_1, \ldots, s_m) \Rightarrow \\
&(s_1 + \cdots + s_m)\mathbf{G}_m = \mathbf{I}_n \otimes \mathbf{G}_m^T vec(\mathbf{A}) + \mathbf{I}_n \otimes \mathbf{R}_{\mathbf{Q}m}^T vec(\mathbf{Q}) + \mathbf{I}_n \otimes \mathbf{G}_{\mathbf{N}m}^T vec(\mathbf{N}) \Rightarrow \\
&(s_1 + \cdots + s_m)\mathbf{G}_m = \begin{bmatrix} \mathbf{I}_n \otimes \mathbf{G}_m^T & \mathbf{I}_n \otimes \mathbf{R}_{\mathbf{Q}m}^T & \mathbf{I}_n \otimes \mathbf{G}_{\mathbf{N}m}^T \end{bmatrix} \begin{bmatrix} vec(\mathbf{A}) \\ vec(\mathbf{Q}) \\ vec(\mathbf{N}) \end{bmatrix}.
\end{aligned}$$

Now, by combining the above two LS problems, we derive for the arbitrary m-tone input, the following mth level least squares problem:

$$\begin{bmatrix} s_1\mathbf{G}_1(s_1) \\ (s_1 + s_2)\mathbf{G}_2(s_1, s_2) \\ \vdots \\ (s_1 + \cdots + s_m)\mathbf{G}_m(s_1, \ldots, s_m) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I}_{n,n} & \mathbf{I}_n \otimes \mathbf{G}_1^T & \mathbf{0}_{n,n^3} & \mathbf{0}_{n,n^2} \\ \mathbf{0}_{n,n} & \mathbf{I}_n \otimes \mathbf{G}_2^T & \mathbf{I}_n \otimes \mathbf{R}_{\mathbf{Q}2}^T & \mathbf{I}_n \otimes \mathbf{R}_{\mathbf{N}2}^T \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{n,n} & \mathbf{I}_n \otimes \mathbf{G}_m^T & \mathbf{I}_n \otimes \mathbf{R}_{\mathbf{Q}m}^T & \mathbf{I}_n \otimes \mathbf{R}_{\mathbf{N}m}^T \end{bmatrix}}_{\mathcal{M}} \begin{bmatrix} vec(\mathbf{B}) \\ vec(\mathbf{A}) \\ vec(\mathbf{Q}) \\ vec(\mathbf{N}) \end{bmatrix}.$$

Remark 5.10 (Quadratic-bilinear identification with m-tone input):
If the rank of the matrix $\mathcal{M}$ is full, $\texttt{rank}(\mathcal{M}) = n^3 + 2n^2 + n$, the identification of the quadratic-bilinear system is possible. The reachability matrices $\mathbf{R_Q}, \mathbf{R_N}$ will provide the information for what m is this possible. ♢

Example 5.1 (Identification of the forced Van der Pol oscillator):
In Example 2.1, we lifted the cubic control system to the quadratic form with state dimension 3. Applying the above algorithm for the parameter $\mu = 0.8$ and solving the least squares problem, we achieve quadratic-bilinear identification with operators that

do not differ numerically from the original.

$$
\begin{aligned}
\mathbf{A}_{id} &= \begin{pmatrix} 3.503e-15 & 1.0 & -6.058e-14 \\ -1.0 & 0.8 & -6.223e-15 \\ -5.209e-15 & 8.035e-15 & 2.813e-15 \end{pmatrix},\ \mathbf{B}_{id} = \begin{pmatrix} -1.571e-15 \\ 1.0 \\ 7.189e-15 \end{pmatrix}, \\
\mathbf{N}_{id} &= \begin{pmatrix} 1.248e-13 & 1.067e-14 & -9.587e-16 \\ 1.563e-14 & -2.337e-14 & -9.875e-16 \\ -4.007e-15 & -3.819e-15 & -1.367e-14 \end{pmatrix}, \\
\mathbf{Q}_{id}^{T} &= \begin{pmatrix} -6.243e-14 & -1.077e-14 & 6.901e-15 \\ 1.04e-14 & 2.16e-14 & 1.0 \\ 9.635e-15 & -1.315e-15 & 7.108e-15 \\ 3.89e-14 & 1.798e-14 & 1.0 \\ -5.067e-14 & -3.604e-14 & 1.828e-15 \\ -2.236e-14 & -0.4 & -2.137e-15 \\ -3.179e-15 & 4.242e-16 & 6.172e-15 \\ 1.949e-14 & -0.4 & -2.7e-15 \\ 6.561e-16 & -3.932e-16 & -2.443e-17 \end{pmatrix}
\end{aligned}
\tag{5.71}
$$

◊

CHAPTER 6

OTHER TYPES OF NONLINEAR SYSTEMS

Contents

6.1. The Hammerstein system

We present a data-driven method for identifying and reducing nonlinear cascaded systems with the Hammerstein structure [103]. The proposed algorithm relies on the Loewner framework (LF), constituting a non-intrusive algorithm for identifying and reducing dynamical systems based on interpolation. We address the problem: the actuator (control input) enters a static nonlinear block. Then, this processed signal is used as an input for a linear time-invariant system (LTI). Additionally, it is considered that the orders of the linear transfer function and the static nonlinearity are not a priori known.

In some engineering applications that study dynamical control systems, the control input enters the differential equations in a nonlinear fashion [159]. It is of interest to identify the hidden nonlinearity while, at the same time, the reduction is needed for robust simulations and control design [7]. The LF [13, 8, 102] constitutes a non-intrusive method that uses only input-output data. The matrix pencil composed of two Loewner matrices reveals the minimality (in terms of McMillan degree) of the LTI system. Using a singular value decomposition (SVD), one can find left and right projection matrices used to construct a low-order model.

The Hammerstein system is characterized by two blocks connected in series, where the static nonlinear (memoryless) block is followed by a linear time-invariant system (LTI) as in Fig. 6.1. The scalar control input-u(t) is used as an argument to the static nonlinearity-$\mathcal{F}$ and then the signal $\mathcal{F}(u(t))$ passes through a linear time-invariant (LTI) system. The static polynomial map approximates other non-polynomial maps (Taylor series expansion) s.a. $\tanh(\cdot)$, $\exp(\cdot)$, etc. The aim is to identify the cascaded system by estimating the coefficients of the polynomial map $k_i,\ i = 1, 2, \ldots, n$ and the hidden LTI system by using only input-output data $(u(t), y(t))$, $t \geq 0$. The steady-state output

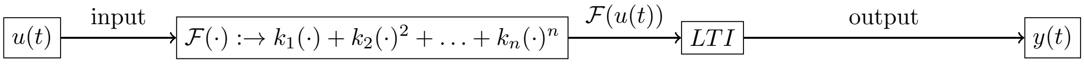

$u(t)$
input
$\mathcal{F}(\cdot) :\to k_1(\cdot) + k_2(\cdot)^2 + \ldots + k_n(\cdot)^n$
$\mathcal{F}(u(t))$
LTI
output
$y(t)$

Figure 6.1.: The input-output scheme of a cascaded system with a static nonlinear (polynomial) map of n^{th} order followed by an LTI. The connection describes a Hammerstein nonlinear model.

solution can be computed explicitly with the convolution integral[1], the impulse response $h(t),\ t \geq 0$ and the linear transfer function $H(j\omega),\ j\omega \in \mathbb{C}$ of the LTI as:

$$\begin{aligned} y(t) &= \left((k_1u(t) + k_2u^2(t) + \ldots + k_nu^n(t)) \star h\right)(t) = \\ &= k_1(u \star h)(t) + k_2(u^2 \star h)(t) + \ldots + k_n(u^n \star h)(t) \\ &= k_1 \int_{-\infty}^{\infty} h(\tau)u(t-\tau)d\tau + \ldots + k_n \int_{-\infty}^{\infty} h(\tau)u^n(t-\tau)d\tau = \\ &= \sum_{i=1}^{n} k_i \int_{-\infty}^{\infty} h(\tau)u^i(t-\tau)d\tau. \end{aligned} \tag{6.1}$$

Let the singleton real input be defined as $u(t) = A\cos(\omega t) = \alpha e^{j\omega t} + \alpha e^{-j\omega t}$ with the amplitude $\alpha = A/2$, the imaginary unit j, the driving frequency $\omega > 0$ and time $t \geq 0$. By substituting the above input in Eq. (6.1) and by making use of the binomial theorem, we conclude that:

$$\begin{aligned} y(t) &= \sum_{i=1}^{n} k_i \int_{-\infty}^{\infty} h(\tau)\left(\alpha e^{j\omega(t-\tau)} + \alpha e^{-j\omega(t-\tau)}\right)^i d\tau = \\ &= \sum_{i=1}^{n}\sum_{m=0}^{i} k_i\alpha^i \frac{i!}{(i-m)!m!} H(j\omega(2m-i))e^{j\omega(2m-i)} \end{aligned} \tag{6.2}$$

At frequency ω, the ℓ^{th} harmonic is computed by applying the single-sided Fourier transform in Eq. (6.2) as:

$$\begin{aligned} Y_{\omega,\ell}(j\ell\omega) &= H(j\ell\omega)\delta(j\ell\omega) \sum_{\ell \leq i \neq 0}^{n} k_i\phi_{i,\ell},\ \ell = 0, \ldots, n, \\ \phi_{i,\ell} &= \begin{cases} 2\alpha^i \cdot {}^iC_{(i+\ell)/2},\ (i \geq \ell) \text{ and } (i+\ell)(even) \\ 0,\ (else) \end{cases},\ \text{where } {}^nC_m = \frac{n!}{(n-m)!m!} \end{aligned} \tag{6.3}$$

[1] $(f \star g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$

6.1.1. The Loewner-Hammerstein identification method

As we have computed the total output of the Hammerstein cascaded system, we proceed with the method of determining the unknowns from input-output data. The symmetry in Eq. (6.3) allows the cancellation of the unknown contribution of the transfer function. Thus, we determine the unknown coefficients k_i and then fit the LTI system using the LF. For this purpose, defining the following invariant frequency quantities $\lambda_{p,q}$ is important.

Definition 6.1 (Frequency invariant quantities):
The $Y_{p,q}$ denotes the q^{th} harmonic at p frequency.

$$\lambda_{p,q} = \frac{Y_{p,q}}{Y_{q,p}} = \frac{\sum_{i=p}^{n} k_i\phi_{i,p}}{\sum_{i=q}^{n} k_i\phi_{i,q}}, p \neq q. \tag{6.4}$$

The entries $\lambda_{p,q}$ are independent of ω. ◊

$$\begin{array}{r|cccccc}
\textit{input} \setminus \textit{harmonic} & 1^{\text{st}} & 2^{\text{nd}} & 3^{\text{rd}} & 4^{\text{th}} & \cdots & n^{\text{th}} \\
\hline
1\omega \rightarrow \boxed{\mathcal{N}} & Y_{1\omega,1} & Y_{1\omega,2} & Y_{1\omega,3} & Y_{1\omega,4} & \cdots & Y_{1\omega,n} \\
2\omega \rightarrow \boxed{\mathcal{N}} & Y_{2\omega,1} & Y_{2\omega,2} & Y_{2\omega,3} & Y_{2\omega,4} & \cdots & Y_{2\omega,n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
n\omega \rightarrow \boxed{\mathcal{N}} & Y_{n\omega,1} & Y_{n\omega,2} & Y_{n\omega,3} & Y_{n\omega,4} & \cdots & Y_{n\omega,n}
\end{array} \tag{6.5}$$

The above harmonic map allows the construction of the following linear system. Due to the mixing linearities (i.e. $k_1 u(t)$ and $(u \star h)(t)$), we can fix k_1 to an arbitrary value. For $p = 1$ and $q = 2, \ldots, n$ results:

$$\begin{bmatrix}
\phi_{21} - \lambda_{12}\phi_{22} & \phi_{31} - \lambda_{12}\phi_{32} & \cdots & \phi_{n1} - \lambda_{12}\phi_{n2} \\
\phi_{21} & \phi_{31} - \lambda_{13}\phi_{32} & \cdots & \phi_{n1} - \lambda_{13}\phi_{n3} \\
\phi_{21} & \phi_{31} & \cdots & \phi_{n1} - \lambda_{14}\phi_{n4} \\
\vdots & \vdots & \ddots & \vdots \\
\phi_{21} & \phi_{31} & \cdots & \phi_{n1} - \lambda_{1n}\phi_{nn}
\end{bmatrix}
\begin{bmatrix} k_2 \\ k_3 \\ k_4 \\ \vdots \\ k_n \end{bmatrix}
= -k_1\phi_{11} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \ \forall k_1 \in \mathbb{R} \setminus \{0\}. \tag{6.6}$$

Finally, as we have identified the scaled (k_1 arbitrary) coefficient vector $\mathbf{k} = (k_1, k_2, \ldots, k_n)$, we can transform the above harmonic map into a measurement map for the linear transfer function as $H(j\ell\omega) = Y_{\omega,\ell}/\sum_{\ell \leq i \neq 0}^{n} k_i\phi_{i,\ell}$. The LTI system is identified and reduced by applying the LF as in [13, 8, 102].

6.1.2. Numerical example

To illustrate the proposed method, we choose the following static nonlinearity as $\mathcal{F}(\cdot) = e^{(\cdot)} - 1$ along with the transfer function $H(j\omega) = 1/((j\omega)^2 + j\omega + 1)$. By exciting the system with $u(t) = 2\cos(\omega_i t)$, $\omega_i = 2\pi[1, 2, \ldots, 10]$ and with collecting the steady state snapshots, we perform Fourier transform for each signal, and we solve the linear system in Eq. (6.6) for $n = 10$ ($Y_{10^{\text{th}}} \sim 1e-10$). The solution is $\hat{\mathbf{k}} = (1, 0.5, 0.167, 0.0417, 0.0083, 0.0014, 1.9676e{-4}, 2.4001e{-5}, 3.1005e{-6}, 3.7401e{-7})$ which constitutes a very good approximation of the leading Taylor series expansion coefficients

Algorithm 6.1: Hammerstein system identification with the Loewner framework

```
Input: Apply signals $u(t) = \alpha\cos(\omega_i t)$ with driving frequencies $\omega_i,\ i = 1, \ldots, n$
       where $n$ is the maximum nonzero harmonic index.
Output: An identified cascaded nonlinear Hammerstein system.
Apply FT and measure $U(j\omega_i)$, $Y_{1^{\text{st}}}(j\omega_i)$, $Y_{2^{\text{nd}}}(2j\omega_i), \ldots, Y_{n^{\text{th}}}(nj\omega_i)$ from the power
    spectrum.
Fix $k_1$ to an arbitrary value and determine the scaled coefficient vector
    $\mathbf{k} = (k_1, k_2, \ldots, k_n)$ by solving the linear system in Eq. (6.6).
Estimate the measurements of the linear transfer function from
    $H(j\ell\omega) = Y_{\omega,\ell} / \sum_{\ell \le i \neq 0}^{n} k_i \phi_{i,\ell}$.
Apply the Loewner framework as in Algorithm 3.1.
```

of the $\mathcal{F}$. Next, we estimate the linear transfer function with the LF [8, 13, 102]. The singular value decay in Section 6.1.2(left) allows the assignment of the order $r = 2$ ($dt = 1e-3$). The transfer function identification is accurate in Section 6.1.2(middle) for the biased $k_1 = 1$ solution. The time domain simulation in Section 6.1.2(right) is independent of the choice of k_1. The large input as $u(t) = 2\text{sawtooth}(0.1\cdot 2\pi t)e^{-0.01t}\cos(0.1\cdot 2\pi t)$ certifies that the method can perform well under large magnitude inputs for nonlinear Hammerstein systems.

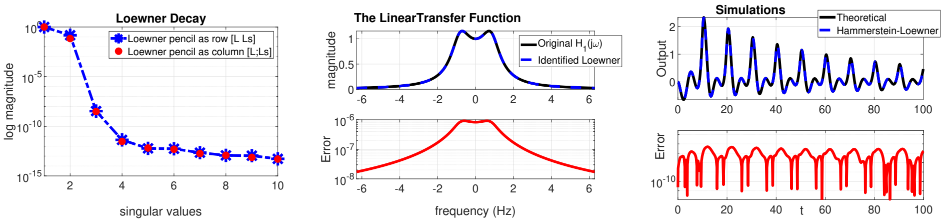


Figure 6.2.: Left: The singular value decay of the Loewner matrices. $\sigma_3/\sigma_1 \sim 1e-10$. Middle: The identified linear transfer function with $\|H - H_r\|_\infty \sim 1e-7$. Right: The simulated identified Hammerstein system in comparison with the original one. $\|y - y_r\|_\infty \sim 1e-7$.

6.2. The Wiener system

In Fig. 6.3, one way of describing a class of non-linear dynamical systems is shown. The control input passes through a linear time-invariant system (LTI) and is scaled by a static nonlinearity. A (SISO) linear time-invariant system is described by the following set of equations:

$$\Sigma_l \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \\ z(t) = \mathbf{C}\mathbf{x}(t), \end{cases} \tag{6.7}$$

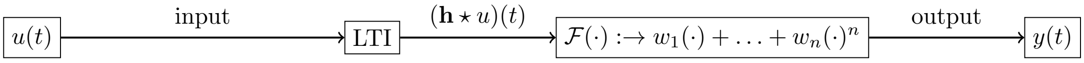

$u(t)$
input
LTI
$(\mathbf{h} \star u)(t)$
$\mathcal{F}(\cdot) :\rightarrow w_1(\cdot) + \ldots + w_n(\cdot)^n$
output
$y(t)$

Figure 6.3.: The Wiener cascaded nonlinear system.

where, $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\mathbf{x}$ (states), z (output), $\mathbf{B}$, $\mathbf{C}^T \in \mathbb{R}^N$. By defining the impulse response (causal kernel) $\mathbf{h}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B}$, $t \geq 0$ and the corresponding transfer function $H(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$, $s \in \mathbb{C}$, we can write the linear output steady state solution in:

Time-domain as:

$$z_s(t) = e^{\mathbf{A}t}\mathbf{z}(0) + \int_{-\infty}^{\infty} \mathbf{h}(\sigma)u(t-\sigma)d\sigma, t \geq 0. \tag{6.8}$$

Frequency-domain (zero initial conditions) as:

$$Z(s) = H(s)U(s), \;\; s \in \mathbb{C}. \tag{6.9}$$

Then, the non-linear output (Wiener) can be written as:

$$y_s(t) = \mathcal{F}(z_s(t)) = \sum_{k=1}^{n} w_k \left(\int_{-\infty}^{\infty} \mathbf{h}(\sigma)u(t-\sigma)d\sigma \right)^k \tag{6.10}$$

Input design: At this point, we need to specify the input. A general input could be the following:

$$u(t) = \sum_{\ell=1}^{L} \alpha_\ell e^{j\omega_\ell t}. \tag{6.11}$$

By substituting this input to the linear solution with zero-initial conditions, we get:

$$\begin{aligned} z_s(t) &= \int_{-\infty}^{\infty} \mathbf{h}(\sigma)u(t-\sigma)d\sigma = \\ &= \int_{-\infty}^{\infty} \mathbf{h}(\sigma) \sum_{\ell=1}^{L} \alpha_\ell e^{j\omega_\ell (t-\sigma)} d\sigma = \\ &= \sum_{\ell=1}^{L} \alpha_\ell e^{j\omega_\ell t} \int_{-\infty}^{\infty} \mathbf{h}(\sigma) e^{-j\omega_\ell \sigma} d\sigma = \\ &= \sum_{\ell=1}^{L} \alpha_\ell H(j\omega_\ell) e^{j\omega_\ell t}. \end{aligned} \tag{6.12}$$

By combining Eq. 6.12 with Eq. 6.10 results to:

$$y_s(t) = \sum_{k=1}^{n} w_k \left(\sum_{\ell=1}^{L} \alpha_\ell H(j\omega_\ell) e^{j\omega_\ell t} \right)^k. \tag{6.13}$$

An 1-tone input: With $L = 2$, constant amplitude α and frequencies as $\omega_1 = \omega$, $\omega_2 = -\omega$ the previous expression remains:

$$\begin{aligned} y_s(t) &= \sum_{k=1}^{n} w_k \left(aH(j\omega)e^{j\omega t} + aH(-j\omega)e^{-j\omega t}\right)^k = \\ &= \sum_{k=1}^{n} w_k a^k \left(H(j\omega)e^{j\omega t} + H(-j\omega)e^{-j\omega t}\right)^k = \\ &= \sum_{k=1}^{n}\sum_{m=0}^{k} w_k a^k \; {}^kC_m H^{k-m}(j\omega)H^m(-j\omega)e^{j\omega(k-2m)t} = \end{aligned} \tag{6.14}$$

As we want to collect the harmonics, we perform the variable change $p = k - 2m \in \mathbb{Z}$.

$$y_s(t) = \sum_{k=1}^{n}\sum_{p=0}^{k} w_k a^k \; {}^kC_{(k-p)/2} H^{(k+p)/2}(j\omega)H^{(k-p)/2}(-j\omega)e^{j\omega pt}. \tag{6.15}$$

So, the p^{th} harmonic is defined as:

$$Y_p(t) = \sum_{\substack{p\leq k\neq 0 \\ (k-p)(\text{even})}}^{n} w_k a^k \; {}^kC_{(k-p)/2} H^{(k+p)/2}(j\omega)H^{(k-p)/2}(-j\omega)e^{j\omega pt} + c.t. \tag{6.16}$$

$$Y_p(pj\omega) = \sum_{\substack{p\leq k\neq 0 \\ (k-p)(\text{even})}}^{n} w_k 2a^k \; {}^kC_{(k-p)/2} H^{(k+p)/2}(j\omega)H^{(k-p)/2}(-j\omega). \tag{6.17}$$

To simplify the notation we introduce the known quantities $\beta_k = 2a^k \; {}^kC_{(k-p)/2}$ and the above equation results as:

$$Y_p(pj\omega) = \sum_{\substack{p\leq k\neq 0 \\ (k-p)(\text{even})}}^{n} w_k \beta_k H^{(k+p)/2}(j\omega)H^{(k-p)/2}(-j\omega). \tag{6.18}$$

Some insight on how to solve this system with unknown ws is to start by fixing the last parameter $w_n = 1$ and get the evaluation of the $H(j\omega) \approx \sqrt[n]{\frac{Y_n}{\beta_n w_n}}$. Then the matrix is known, and by solving the system, we get the remaining $(n-1)$ coefficients ws.

Remark 6.2 (On the identification of Wiener systems):
The last harmonic is usually corrupted by numerical noise. So, the $H(j\omega)$ evaluation depends on that. This could lead to inaccurate identification of the Wiener system. ◊

A 2-tone complex input: With $L = 2$, a flat amplitude α and frequencies as ω_1, ω_2, we have:

$$\begin{aligned} y_s(t) &= \sum_{k=1}^{n} w_k \left(aH(j\omega_1)e^{j\omega_1 t} + aH(j\omega_2)e^{j\omega_2 t}\right)^k = \\ &= \sum_{k=1}^{n} w_k a^k \left(H(j\omega_1)e^{j\omega_1 t} + H(j\omega_2)e^{j\omega_2 t}\right)^k = \\ &= \sum_{k=1}^{n}\sum_{m=0}^{k} w_k a^k \, {}^kC_m H^{k-m}(j\omega_1)H^m(j\omega_2)e^{j[k\omega_1+m(\omega_2-\omega_1)]t} = \end{aligned} \tag{6.19}$$

An n-tone complex input

$$\begin{aligned} y_s(t) &= \sum_{k=1}^{n} w_k \left(\sum_{i=1}^{L} \alpha_i H(j\omega_i)e^{j\omega_i t}\right)^k = \\ &= \sum_{k=1}^{n} w_k \left(a_1 H(j\omega_1)e^{j\omega_1 t} + a_2 H(j\omega_2)e^{j\omega_2 t} + \ldots + a_L H(j\omega_L)e^{j\omega_L t}\right)^k \end{aligned} \tag{6.20}$$

To simplify the notation above, we introduce $\beta_i = a_i H(j\omega_i),\ i = 1, \ldots, L$.

$$\begin{aligned} y_s(t) &= \sum_{k=1}^{n} w_k \left(\sum_{i=1}^{L} \beta_i e^{j\omega_i t}\right)^k = \sum_{k=1}^{n} w_k \left(\beta_1 e^{j\omega_1 t} + \beta_2 e^{j\omega_2 t} + \ldots + \beta_L e^{j\omega_L t}\right)^k \\ &= \sum_{k=1}^{n} \sum_{m_1+\ldots+m_L=k} w_k \frac{k!}{m_1!m_2!\cdots m_L!} \prod_{q=1}^{L} \beta_q^{m_q} e^{j\omega_q m_q t} \\ &= \sum_{k=1}^{n} \sum_{m_1+\ldots+m_L=k} w_k \frac{k!}{m_1!m_2!\cdots m_L!} \beta_1^{m_1}\cdots\beta_L^{m_L} e^{j(\omega_1 m_1+\ldots+\omega_L m_L)t}, \end{aligned} \tag{6.21}$$

The above expression indicates the construction of $\begin{pmatrix} n+L-1 \\ L-1 \end{pmatrix}$ harmonics. We can obtain different goals by choosing the frequencies $\omega_q,\ q = 1, \ldots, L$ with different strategies, e.g., real input signal and harmonics separation. A study for various cases is presented.

6.2.1. The Wiener-Loewner identification algorithm

6.2.2. Numerical examples

$L = 4$ **with two pairs and** $n = 2$**.** We start with $L = 4$ complex exponential terms. The frequencies are: $\omega_q,\ q = 1, 2, 3, 4$. By choosing pairs like $\omega_3 = -\omega_1$ and $\omega_4 = -\omega_2$, with a flat magnitude $\alpha_q = 1$, we can write the 2-harmonic input as a real-valued quantity:

$$u(t) = e^{j\omega_1} + e^{-j\omega_1} + e^{j\omega_2} + e^{-j\omega_2} = 2\cos(\omega_1 t) + 2\cos(\omega_2 t),\ \omega_1, \omega_2 > 0,\ t \geq 0. \tag{6.22}$$

Algorithm 6.2: Wiener system identification with the Loewner framework

```
Input: Apply multi-tone harmonic signals
Output: An identified cascaded nonlinear Wiener system.
1: Apply FT and measure the propagating harmonics from the power spectrum.
2: Determine the polynomial coefficients $\omega_1, \omega_2, ...$ from the power spectrum as
   explained in Section 6.2.2.
3: Get the estimations of the linear transfer function.
4: Apply the linear Loewner framework Algorithm 3.1.
```

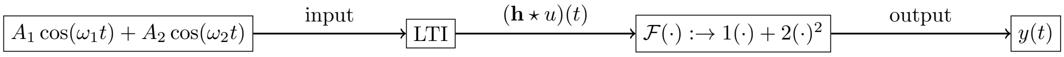

$A_1\cos(\omega_1 t) + A_2\cos(\omega_2 t)$
input
LTI
$(\mathbf{h} \star u)(t)$
$\mathcal{F}(\cdot) :\to 1(\cdot) + 2(\cdot)^2$
output
$y(t)$

Figure 6.4.: The input-output scheme of a cascaded LTI with a static (memoryless) nonlinear polynomial map of order 2.

With $\phi^{(k)}_{m_1,m_2,m_3,m_4} = \frac{k!}{m_1!m_2!m_3!m_4!}\beta_1^{m_1}\beta_2^{m_2}\beta_3^{m_3}\beta_4^{m_4}$ and $n = 2$ the steady state output is the following:

$$\begin{aligned} y_s(t) = \sum_{k=1}^{2} \sum_{m_1+m_2+m_3+m_4=k} & w_k \phi^{(k)}_{m_1,m_2,m_3,m_4} e^{j(\omega_1(m_1-m_3)+\omega_2(m_2-m_4))t} = \\ &= w_1(\phi^{(1)}_{1000}e^{j\omega_1 t} + \phi^{(1)}_{0100}e^{j\omega_2 t} + \phi^{(1)}_{0010}e^{-j\omega_1 t} + \phi^{(1)}_{0001}e^{-j\omega_2 t}) + \\ &+ w_2(\phi^{(2)}_{2000}e^{2j\omega_1 t} + \phi^{(2)}_{0200}e^{2j\omega_2 t} + \phi^{(2)}_{0020}e^{-2j\omega_1 t} + \phi^{(2)}_{0002}e^{-2j\omega_2 t} + \\ &+ \phi^{(2)}_{1100}e^{j(\omega_1+\omega_2)t} + \phi^{(2)}_{1010}e^{j(0)t} + \phi^{(2)}_{1001}e^{j(\omega_1-\omega_2)t} + \phi^{(2)}_{0110}e^{(-\omega_1+\omega_2)t} \\ &+ \phi^{(2)}_{0101}e^{-j(0)t} + \phi^{(2)}_{0011}e^{j(-\omega_1-\omega_2)t}). \end{aligned} \tag{6.23}$$

The above expression contains 14 terms. By collecting those, we have up to 14/2=7 distinct harmonics.

Time Domain:

$$\begin{aligned} \text{DC} : Y_{0\omega_1+0\omega_2}(t) &= w_2(\phi^{(2)}_{1010}e^{j(0)t} + \phi^{(2)}_{0101}e^{-j(0)t}) \\ \text{"1-0"} : Y_{1\omega_1+0\omega_2}(t) &= w_1(\phi^{(1)}_{1000}e^{j\omega_1 t} + \phi^{(1)}_{0010}e^{-j\omega_1 t}) \\ \text{"0-1"} : Y_{0\omega_1+1\omega_2}(t) &= w_1(\phi^{(1)}_{0100}e^{j\omega_2 t} + \phi^{(1)}_{0001}e^{-j\omega_2 t}) \\ \text{"1-1"} : Y_{1\omega_1+1\omega_2}(t) &= w_2(\phi^{(2)}_{1100}e^{j(\omega_1+\omega_2)t} + \phi^{(2)}_{0011}e^{-j(\omega_1+\omega_2)t}) \\ \text{"1-1"} : Y_{1\omega_1-1\omega_2}(t) &= w_2(\phi^{(2)}_{1001}e^{j(\omega_1-\omega_2)t} + \phi^{(2)}_{0110}e^{-j(\omega_1-\omega_2)t}) \\ \text{"2-0"} : Y_{2\omega_1+0\omega_2}(t) &= w_2(\phi^{(2)}_{2000}e^{2j\omega_1 t} + \phi^{(2)}_{0020}e^{-2j\omega_1 t}) \\ \text{"0-2"} : Y_{0\omega_1+2\omega_2}(t) &= w_2(\phi^{(2)}_{0200}e^{2j\omega_2 t} + \phi^{(2)}_{0002}e^{-2j\omega_2 t}), \end{aligned} \tag{6.24}$$

where the general harmonic index with two real harmonic input frequencies ($L = 4$) is

denoted by: $Y_{\lambda_1\omega_1+\lambda_2\omega_2}(t),\ \lambda_1,\lambda_2 \in \{0,1,\dots,n\}$ with $\lambda_1+\lambda_2 \leq n$.
Frequency Domain:

$$\begin{aligned}
\text{DC} : Y_{0\omega_1+0\omega_2} &= w_2(\phi^{(2)}_{1010}+\phi^{(2)}_{0101}) \\
\text{"1-0"} : Y_{1\omega_1+0\omega_2} &= w_1\phi^{(1)}_{1000} \\
\text{"0-1"} : Y_{0\omega_1+1\omega_2} &= w_1\phi^{(1)}_{0100} \\
\text{"1-1"} : Y_{1\omega_1+1\omega_2} &= w_2\phi^{(2)}_{1100} \\
\text{"1-1"} : Y_{1\omega_1-1\omega_2} &= w_2\phi^{(2)}_{0110} \\
\text{"2-0"} : Y_{2\omega_1+0\omega_2} &= w_2\phi^{(2)}_{2000} \\
\text{"0-2"} : Y_{0\omega_1+2\omega_2} &= w_2\phi^{(2)}_{0200},
\end{aligned} \tag{6.25}$$

At this step, the choices of ω_1 and ω_2 are crucial for the harmonic separation. In the power spectrum, the harmonics to appear are given next:

$$\{0, \omega_2-\omega_1, \omega_1, \omega_2, \omega_1+\omega_2, 2\omega_1, 2\omega_2\} \tag{6.26}$$

Fig. 6.5 shows one choice of ω_2 where the whole harmonic spectrum remains well separated. By choosing $\omega_2 = \phi\omega_1$, with $\phi = 1.5$, we obtain a full separation scheme. Fig. 6.5(left) is the theoretical interpretation of the separation scheme. Next is the power spectrum Fig. 6.5(right) of the Wiener process in a steady state.

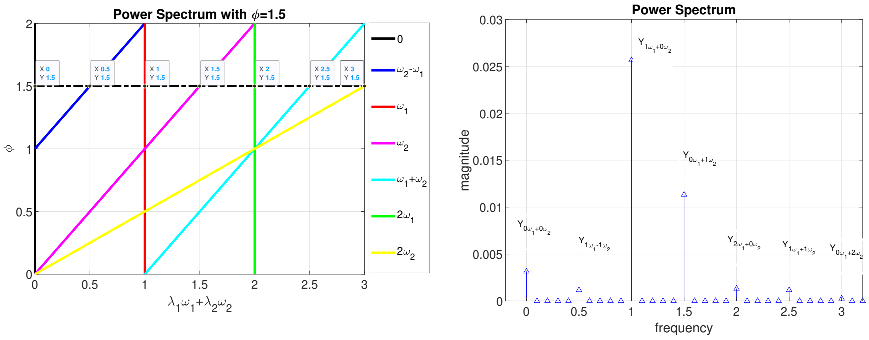


Figure 6.5.: Left: Identification of a quadratic Wiener system. A two-harmonic input passes through a linear time-invariant system (LTI) and scales from a quadratic non-linearity. Right: Frequency spectrum and harmonic indexing.

	$Y_{0\omega_1+0\omega_2}$	$Y_{1\omega_1+1\omega_2}$	$Y_{1\omega_1-1\omega_2}$	$Y_{2\omega_1+0\omega_2}$	$Y_{0\omega_1+2\omega_2}$
$\hat{w}_2$	2.0	$2.0-4.5166e-8\mathrm{i}$	$2.0-1.1955e-7\mathrm{i}$	$2.0-4.4502e-8\mathrm{i}$	$2.0-1.4207e-7\mathrm{i}$

Table 6.1.: Inferred harmonics from the power spectrum and indexing

By putting together those measurements, we can proceed as follows:

$$\begin{aligned}
Y_{0\omega_1+0\omega_2} &= w_2(\phi^{(2)}_{1010}+\phi^{(2)}_{0101}) = w_2(2|H(j\omega_1)|^2+2|H(j\omega_2)|^2)\\
Y_{1\omega_1+0\omega_2} &= w_1\phi^{(1)}_{1000} = w_1 H(j\omega_1)\\
Y_{0\omega_1+1\omega_2} &= w_1\phi^{(1)}_{0100} = w_1 H(j\omega_2)\\
Y_{1\omega_1+1\omega_2} &= w_2\phi^{(2)}_{1100} = w_2 2H(j\omega_1)H(j\omega_2)\\
Y_{1\omega_1-1\omega_2} &= w_2\phi^{(2)}_{0110} = w_2 2H(j\omega_2)H(-j\omega_1)\\
Y_{2\omega_1+0\omega_2} &= w_2\phi^{(2)}_{2000} = w_2 H^2(j\omega_1)\\
Y_{0\omega_1+2\omega_2} &= w_2\phi^{(2)}_{0200} = w_2 H^2(j\omega_2),
\end{aligned}\tag{6.27}$$

Remark 6.3 (Mixed linearities):
For the above cascaded Wiener system, we assume that when the underlying system is purely linear, the input signal should be scaled only from the LTI. The memoryless box wouldn't play any role in it. Then, a logical assumption without restriction is to set $w_1 = 1$. ◊

After the above remark, the new goal is to determine the parameter w_2 and the linear transfer function H. From the Fig. 6.3, we have:

- $Y_{1\omega_1+0\omega_2} = -0.025274 - 0.0042922\mathrm{i}$,
- $Y_{0\omega_1+1\omega_2} = -0.011239 - 0.0013139\mathrm{i}$.

Then from Eq. 6.27, with $w_1 = 1$ we get the two evaluation of the transfer function at $\omega_1 = 1$ and $\omega_2 = 1.5$.

$$\begin{aligned}
H(j\omega_1) &\approx \hat{H}(j\omega_1) = -0.025274 - 0.0042922\mathrm{i},\\
H(j\omega_2) &\approx \hat{H}(j\omega_2) = -0.011239 - 0.0013139\mathrm{i}.
\end{aligned}\tag{6.28}$$

From the set of Eqs. 6.27, we get the evaluation of w_2.

An evaluation of w_2 is the following:

$$w_2 \approx \frac{\sum_{i=1}^{5}\hat{w_2}^{(i)}}{5} = 2.0 - 7.0258e - 8\mathrm{i}. \tag{6.29}$$

We get $w_2 = 1.999999849625013$ by enforcing real values. We have identified the parameters of the static non-linearity of order-2 ($w_1 = 1$ and $w_2 = 1.999999849625013$). We use the Loewner framework Algorithm 3.1 to identify the transfer function.

Finally, the simulation depicting the responses of the original and the identified systems is provided in Fig. 6.6.

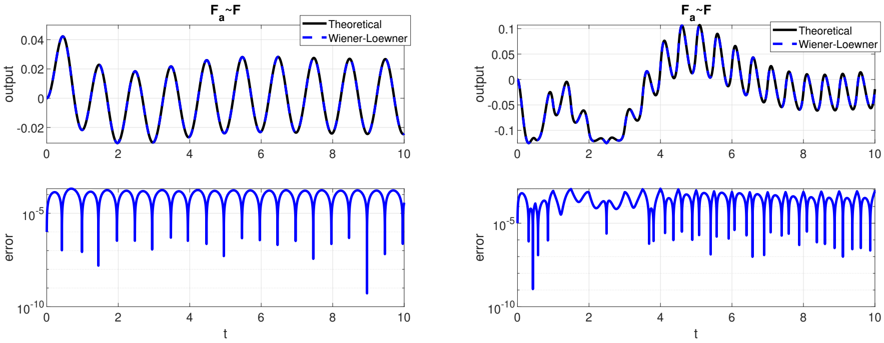


Figure 6.6.: The simulation error between the original $\mathbf{F}$ and the identified Wiener-Loewner model $\mathbf{F}_a$. Left: the input is the probing one used for the measurements. Right: The input is a $\texttt{sawtooth}(t)$ with a big amplitude ($a = 10$). A uniformly distributed error profile of the order $O(10^{-5})$ is depicted in both cases.

Example 6.1 (A nonlinear circuit):
We start with $L = 6$ complex exponential terms. The frequencies are: $\omega_q,\ q = 1, 2, 3, 4, 5, 6$. By choosing pairs like $\omega_4 = -\omega_1$ and $\omega_5 = -\omega_2$ and $\omega_6 = -\omega_3$, with magnitudes a_1, a_2, a_3, we can write the 3-harmonic input as real:

$$\begin{aligned} u(t) &= a_1 e^{j\omega_1 t} + a_1 e^{-j\omega_1 t} + a_2 e^{j\omega_2 t} + a_2 e^{-j\omega_2 t} + a_3 e^{j\omega_3 t} + a_3 e^{-j\omega_3 t} \\ &= 2a_1 \cos(\omega_1 t) + 2a_2 \cos(\omega_2 t) + 2a_3 \cos(\omega_3 t),\ \ \omega_1, \omega_2, \omega_3 > 0,\ \ t \geq 0. \end{aligned} \tag{6.30}$$

Next is the steady state output for the Wiener system with L complex input and nth order polynomial static non-linearity. For simplicity the notation $\beta_i = a_i H(j\omega_i),\ i = 1, \ldots, L$ is introduced.

$$y_{ss}(t) = \sum_{k=1}^{n} \sum_{m_1+\ldots+m_L=k} w_k \frac{k!}{m_1! m_2! \cdots m_L!} \beta_1^{m_1} \cdots \beta_L^{m_L} e^{j(\omega_1 m_1 + \ldots + \omega_L m_L)t}. \tag{6.31}$$

By fixing $L = 6$ and with the above "real" signal symmetry, we can rewrite

$$\begin{aligned} y_{ss}(t) &= \sum_{k=1}^{n} \sum_{m_1+\ldots+m_6=k} w_k \frac{k!}{m_1! m_2! \cdots m_6!} \beta_1^{m_1} \cdots \beta_6^{m_6} e^{j(\omega_1 m_1 + \ldots + \omega_6 m_6)t} \\ &= \sum_{k=1}^{n} \sum_{m_1+\ldots+m_6=k} w_k \frac{k!}{m_1! m_2! \cdots m_6!} \beta_1^{m_1} \cdots \beta_6^{m_6} e^{j(\omega_1 m_1 + \ldots + \omega_6 m_6)t} \\ &= \sum_{k=1}^{n} \sum_{m_1+\ldots+m_6=k} w_k \frac{k!}{m_1! m_2! \cdots m_6!} \beta_1^{m_1} \cdots \beta_6^{m_6} e^{j(\omega_1 (m_1 - m_2) + \omega_2 (m_3 - m_4) + \omega_3 (m_5 - m_6))t}. \end{aligned}$$

By denoting $m_1 - m_2 = k_1,\ m_4 - m_3 = k_2,\ m_6 - m_5 = k_3$

$$
\begin{aligned}
&Y(k_1\omega_1 + k_2\omega_2 + k_3\omega_3) = \\
&= \sum_{k=1}^{n} \sum_{k_1+k_2+k_3+2(m_2+m_4+m_6)=k} w_k \frac{k!}{(k_1+m_2)!m_2!(k_2+m_4)!m_4!(k_3+m_6)!m_6!} \cdot \\
&\cdot \beta_1^{k_1+m_2} \beta_2^{m_2} \beta_3^{k_2+m_4} \beta_4^{m_4} \beta_5^{k_3+m_6} \beta_6^{m_6} \\
&= \sum_{k=1}^{n} \sum_{k_1+k_2+k_3+2(m_2+m_4+m_6)=k} w_k \frac{k!}{(k_1+m_2)!m_2!(k_2+m_4)!m_4!(k_3+m_6)!m_6!} \cdot \\
&\cdot a_1^{k_1+2m_2} a_2^{k_2+2m_4} a_3^{k_3+2m_6} H^{k_1+m_2}(j\omega_1) H^{m_2}(-j\omega_1) H^{k_2+m_4}(j\omega_2) H^{m_4}(-j\omega_2) H^{k_3+m_6}(j\omega_3) H^{m_6}(-j\omega_3)
\end{aligned}
$$

Application $Y(\omega_3)$. We set $k_1 = k_2 = 0,\ k_3 = 1$. Then

$$
\begin{aligned}
Y(\omega_3) = &\sum_{k=1}^{n} \sum_{1+2(m_2+m_4+m_6)=k} w_k \frac{k!}{m_2!m_2!m_4!m_4!(1+m_6)!m_6!} \cdot \\
&\cdot a_1^{2m_2} a_2^{2m_4} a_3^{1+2m_6} H^{m_2}(j\omega_1) H^{m_2}(-j\omega_1) H^{m_4}(j\omega_2) H^{m_4}(-j\omega_2) H^{1+m_6}(j\omega_3) H^{m_6}(-j\omega_3)
\end{aligned}
\tag{6.32}
$$

And now we expand (up to $k = 3$ as we have assumed cubic nonlinearity)

$$
\begin{aligned}
Y(\omega_3) = &\overbrace{w_1 a_3 H(j\omega_3)}^{k_1} + \overbrace{6w_3 a_1^2 a_3 H(j\omega_1) H(-j\omega_1) H(j\omega_3)}^{k_3} + \\
&+ \underbrace{6w_3 a_2^2 a_3 H(j\omega_2) H(-j\omega_2) H(j\omega_3) + 3w_3 a_3^3 H(j\omega_3) H(j\omega_3) H(-j\omega_3)}_{k_3}
\end{aligned}
\tag{6.33}
$$

By denoting $H_1(s_1) = w_1 H(s_1)$, $H_2(s_1, s_2) = w_3 H(s_1) H(s_2)$, and $H_3(s_1, s_2, s_3) = w_3 H(s_1) H(s_2) H(s_3)$, we result to the more general **Volterra⊃Wiener** expression as

$$
\begin{aligned}
Y(\omega_3) = &\, a_3 H_1(j\omega_3) + 6a_1^2 a_3 H_3(j\omega_1, -j\omega_1, j\omega_3) + \\
&+ 6a_2^2 a_3 H_3(j\omega_2, -j\omega_2, j\omega_3) + 3a_3^3 H_3(j\omega_3, j\omega_3, j\omega_3).
\end{aligned}
\tag{6.34}
$$

Measurements: Now, the task is to get the measurement $Y(\omega_3)$.

1. Taking care of the overlapping harmonics by choosing appropriately ω_1, ω_2, ω_3 and waiting for steady state, from the power spectrum (FT), we can get the measurements.

2. Using the Agilent technology, X-parameters generation as described in [153, 150] with a large signal consisting of 3-tones will yield the value of $Y(\omega_3)$.

One practical choice for separating the harmonics (kernels) is the following. You take ω_1 and the choice of ω_2 at least nw_1 far with n to be the maximum harmonic index that you want to involve. The same can be assumed for the ω_3. To exploit the second kernel as well, we have:

$$
Y(2\omega_1) = w_2 a_2^2 H(j\omega_2) H(j\omega_2) = a_2^2 H_2(j\omega_2, j\omega_2). \tag{6.35}
$$

Kernel separation with amplitude shifting From Eq. (6.34), by varying the amplitudes, we can solve the following least square problem to separate the kernels:

$$\begin{bmatrix} a_3^{(1)} & 6a_1^2 a_3^{(1)} & 6a_2^2 a_3^{(1)} & 3(a_3^3)^{(1)} \\ a_3^{(2)} & 6a_1^2 a_3^{(2)} & 6a_2^2 a_3^{(2)} & 3(a_3^3)^{(2)} \\ a_3^{(3)} & 6a_1^2 a_3^{(3)} & 6a_2^2 a_3^{(3)} & 3(a_3^3)^{(3)} \\ a_3^{(4)} & 6a_1^2 a_3^{(4)} & 6a_2^2 a_3^{(4)} & 3(a_3^3)^{(4)} \end{bmatrix} \cdot \begin{bmatrix} H_1(j\omega_3) \\ H_3(j\omega_1, -j\omega_1, j\omega_3) \\ H_3(j\omega_2, -j\omega_2, j\omega_3) \\ H_3(j\omega_3, j\omega_3, j\omega_3) \end{bmatrix} = \begin{bmatrix} Y(\omega_3)^{(1)} \\ Y(\omega_3)^{(2)} \\ Y(\omega_3)^{(3)} \\ Y(\omega_3)^{(4)} \end{bmatrix} \tag{6.36}$$

We quote the following setup similarly to in [153]. The linear transfer function is $H(s) = 1/(LCs^2 + RCs + 1)$ with $L = 42.52,\ C = 8.5 \cdot 10^{-3},\ R = 50$. If we denote the linear output $y_l(t)$, then, the non-linear output is given along with the 3rd-order Taylor expansion:

$$y(t) = \frac{1}{(8 - y_l)} \left(4y_l(t) + 2y_l^2(t) + y_l^3(t)\right) \approx \frac{1}{2} y_l(t) + \frac{5}{16} y_l^2(t) + \frac{21}{128} y_l^3(t). \tag{6.37}$$

The goal is to identify the above system and test it under a rectangular input.

Measurements: We train the system with a 3-tone input with the following parameters:

- Amplitudes: $a_1 = a_2 = 0.1$ and $a_3 = [0.1\ 0.2\ 0.3\ 0.4]$.
- Frequencies: $\omega_1 = 0.1,\ \omega_2 = 0.5,\ \omega_3 = 2.$

$$\underbrace{\begin{bmatrix} 0.2 & 0.012 & 0.012 & 0.024 \\ 0.3 & 0.018 & 0.018 & 0.081 \\ 0.4 & 0.024 & 0.024 & 0.192 \\ 0.5 & 0.03 & 0.03 & 0.375 \end{bmatrix}}_{rank=2} \cdot \begin{bmatrix} H_1(j\omega_3) \\ H_3(j\omega_1, -j\omega_1, j\omega_3) \\ H_3(j\omega_2, -j\omega_2, j\omega_3) \\ H_3(j\omega_3, j\omega_3, j\omega_3) \end{bmatrix} = \begin{bmatrix} -0.0018116 - 0.00019742\text{i} \\ -0.0027174 - 0.00029528\text{i} \\ -0.0036232 - 0.00039315\text{i} \\ -0.004529 - 0.00049103\text{i} \end{bmatrix} \tag{6.38}$$

$$\begin{array}{l|c} \text{Kernel} & \text{measurement} \\ H_1(j\omega_3) & -0.0089931 - 0.00097902\text{i} \\ H_3(j\omega_1, -j\omega_1, j\omega_3) & -0.00053959 - 0.000058741\text{i} \\ H_3(j\omega_2, -j\omega_2, j\omega_3) & -0.00053959 - 0.000058741\text{i} \\ H_3(j\omega_3, j\omega_3, j\omega_3) & -1.0245e - 7 + 5.9452e - 6\text{i} \end{array} \tag{6.39}$$

Sweeping ω_3, we get more measurements.So, $H(3j2\pi) = -0.0039777 - 0.0003308\text{i}$. Now, we will construct the Loewner model with these 2-measurements. The identify linear transfer function $\hat{H}_1(j\omega) = \mathbf{C}_{\text{id}} \left(j\omega \mathbf{I} - \mathbf{A}_{\text{id}}\right)^{-1} \mathbf{C}_{\text{id}}$ has the following system

$$\mathbf{A}_{\text{id}} = \begin{bmatrix} -1.1766 & -18.85 \\ 0.14796 & 1.1345e - 16 \end{bmatrix}, \ \mathbf{B}_{\text{id}} = \begin{bmatrix} -0.83196 \\ -13.224 \end{bmatrix}, \ \mathbf{C}_{\text{id}} = \begin{bmatrix} -0.0056 & 0.0005 \end{bmatrix}. \tag{6.40}$$

It remains

$$H_3(j\omega_1, -j\omega_1, j\omega_3) = w_3 H(j\omega_1) H(-j\omega_1) H(j\omega_3) \approx w_3 \hat{H}_1(j\omega_1) \hat{H}_1(-j\omega_1) \hat{H}_1(j\omega_3)/w_1^3,$$

and this leads to

$$w_3/w_1^3 = 0.18961 + 4.7066e - 18\text{i}.$$

Similar,

$$H_2(j\omega_2, j\omega_2) \approx w_2 \hat{H}_1(j\omega_2) \hat{H}_1(j\omega_2)/w_1^2,$$

so,

$$w_2/w_1^2 = 0.3135 + 0.0058578\text{i}.$$

◊

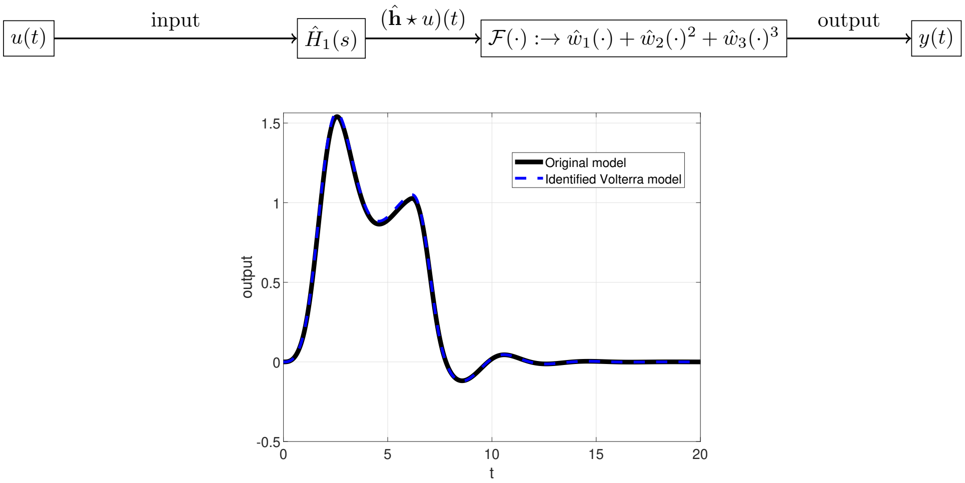


Figure 6.7.: The identified cascaded Wiener nonlinear system in comparison with the original [153].

Remark 6.4 (Identification of Wiener-Hammerstein systems):
Identifying the cascaded Hammerstein-Wiener system could be possible if we can identify a multi-tone input when it passes through a static polynomial non-linearity. This happens because the input in the Wiener branch is multi-tone, as it comes from the Hammerstein branch. ◊

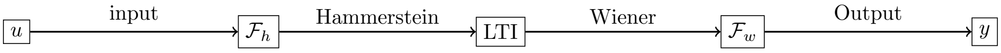


Figure 6.8.: The cascaded Hammerstein-Wiener system.

CHAPTER 7 CONCLUSIONS

Contents

7.1. Summary

In this thesis, we were concerned with constructing nonlinear dynamical models from input-output time-domain data. The developed methods use the Loewner framework to realize the linear subsystem, and continuously certain nonlinear operators (bilinear, quadratic-bilinear, Hammerstein, Wiener) were fitted to improve the overall performance. The ultimate goal was to devise methods that achieved two aims. The first aim was to realize the physical data (time domain) for model discovery and for serving the scope of identification under a finite assumed numerical precision and model structure. The second aim was to employ model-reduction techniques to construct low-order dynamical systems for robust simulation. Within the thesis, we have analyzed theoretically under which assumptions the developed methods achieve the two aims in several examples and applications.

7.2. Outlook

In Chapter 1, we motivated the research direction for data-driven modeling that stays consistent with the formal mathematical description of the physical laws. We emphasized the importance of constructing interpretable models, i.e., state-space dynamical systems, due to their reliability which is essential for engineering applications. Also, we motivated the excellent characteristics of the data learning methods that would benefit if these two approaches (data science and data engineering) could be mutually integrated.

In Chapter 2, we introduced some essential linear algebra tools and their properties that we built our methods.

In Chapter 3, we reviewed the applicability of the Loewner framework to several linear time-invariant dynamical systems. Mainly, we emphasized technical issues and gave insights to make this method easy to use and avoid pitfalls. We investigated the rational approximation to non-rational transfer functions which provided good models capable of approximating the dynamics of infinite dimensional linear systems within the interpolation regime.

In Chapter 4, we started our analysis with the continuous in-time bilinear control systems, and an algorithm that constructs bilinear models from time-domain harmonic data was introduced. Examples that illustrated the identification or the reduction goals were detailed. Connection with the developed bilinear Loewner has been achieved when time-domain data have been considered. Although these developed methods directly applied to the continuous in-time bilinear control systems case, we also investigated the discrete in-time case, and connections with the bilinear realization theory and the subspace method were included. All the identification methods (e.g., subspace) for bilinear systems either need more expensive, repetitive simulations or exponential computational complexity. To relax this issue, we utilize machine learning techniques, e.g., Neural Networks, due to their power to learn input-output maps through the universal approximation theorem. In a real application example, we achieved interpretable bilinear models that explained the nonlinear data with comparable performance with the subspace method. We considered this result a successful connection between data engineering and learning methods.

In Chapter 5, we introduced data-driven modeling with the continuous in-time quadratic-bilinear control systems. This class of systems has been very well studied in recent years. Higher symmetric Volterra kernels should be included to realize the quadratic operator from input-output harmonic data. The proposed method uses data from the first three symmetric Volterra kernels to realize the quadratic system. Nonlinear algebraic equations must be solved. After using measurements from the third kernel, we had to solve a quadratic vector equation to upgrade the missing information from the null space parametrization of the second kernel. That was successful after utilizing an iterative (fixed point) algorithm based on Newton's method. We illustrated this identification result with the forced Lorenz attractor. In particular, when the parameters of the forced Lorenz attractor produced two non-trivial equilibrium points, we could identify the global quadratic system that bifurcated to its nonzero equilibrium points after aligning the resulting invariant operators. The algorithm that aligns the two quadratic systems without the access of the linear operators involves the solution of a constrained quadratic matrix equation. We solved this matrix equation by utilizing Newton's method in a similar way that Kleinman's algorithm solves the Ricatti equation with Fréchet differentiability. Harmonic time-domain data that can be processed in the Fourier spectrum and for extracting estimations of the symmetric Volterra kernels were tested for a larger scale example, and 98% reduction performance with five-digit accuracy was also reported. A similar algorithm was introduced for constructing quadratic-bilinear state-space systems from harmonic data with applications in electrical engineering. The proposed methods upgrade the estimation of the operators by involving higher kernels after parameterizing the nonempty null space. Involving even higher Volterra kernels leads to solving higher polynomial vector equations with the tensor formulation, which seems numerical challenging and is left for future research endeavors.

In Chapter 6, we extended the Loewner framework to specific structures of cascaded nonlinear systems known as Hammerstein-Wiener. We provided the Loewner-Hammerstein algorithm for realizing such systems. We started with the Hammerstein system that interprets the input that inserts in the LTI nonlinearly. Continuously, and for the same reason, we introduced a similar algorithm for the Wiener system that interprets the output from the LTI nonlinearly. The algorithms were detailed with application examples and insight into combining the Loewner framework with the Hammerstein-Wiener structure was given.

Appendices

APPENDIX A

APPENDIXA

The Probing method. It was shown by Rugh [137], and Billings [35] that for nonlinear systems which are described by the Volterra model Eq. (2.28) and excited by a combination of complex exponentials $u(t) = \sum_{i=1}^{R} e^{s_i t}, \quad 1 \leq R \leq N$, the output response can be written as

$$\begin{aligned} y(t) &= \sum_{n=1}^{N} \sum_{i_1=1}^{R} \cdots \sum_{i_n=1}^{R} H_n(s_{i_1}, \ldots, s_{i_n}) e^{(s_{i_1} + \cdots + s_{i_n})t}, \\ &= \sum_{n=1}^{N} \sum_{m(n)} \tilde{H}_{m_1(n) \cdots m_R(n)}(s_1, \ldots, s_R) e^{(m_1(n)s_1 + \cdots + m_R(n)s_R)t}, \end{aligned} \tag{A.1}$$

where $\sum_{m(n)}$ indicates an R-fold sum over all integer indices $m_1(n), \ldots, m_R(n)$ such that $0 \leq m_i(n) \leq n, \;\; m_1(n) + \cdots + m_R(n) = n$, and

$$\tilde{H}_{m_1(n) \cdots m_R(n)}(s_1, \ldots, s_R) = \frac{n!}{m_1(n)! \cdots m_R(n!)} H_n(\underbrace{s_1, \ldots, s_1}_{m_1(n)}, \ldots, \underbrace{s_R, \ldots, s_R}_{m_R(n)}). \tag{A.2}$$

Note that, $\tilde{H}_n$ is the weighted GFRF, corresponding to H_n; The former scales with the factor $\frac{n!}{m_1(n)! \cdots m_R(n!)}$. Note also that different input amplitudes can be considered as in [154, 35] where amplitude shifting allows kernel separation.

To determine the Rth-order generalized frequency response function $H_R(s_1, \ldots, s_R)$, the probing input $u(t) = \sum_{i=1}^{R} e^{s_i t}$ needs to be applied, with at least R harmonics. We introduce the input-state GFRFs $\mathbf{G}_i(s_1, \ldots, s_i), \; i = 1, ..., n$ to simplify the next derivations. These simply result in the input-output GFRFs by multiplying the $\mathbf{H}_i$'s from the left with the output vector $\mathbf{C}$ (in the SISO case), i.e., $H_n(s_1, \ldots, s_n) = \mathbf{C}\mathbf{G}_n(s_1, \ldots, s_n)$. Note further that the transfer function $\mathbf{G}_i$ is a vector of length equal to the state dimension n (this latter is identical to $\mathbf{H}_i$ when $\mathbf{C} = \mathbf{I}_n$, i.e., when all the state elements are individually observed).

- **$R = 1$ - 1st order GFRF $H_1(s_1)$:** With input $u(t) = e^{s_1 t}$ the state solution $\mathbf{x}(t)$ and the time derivative $\dot{\mathbf{x}}(t)$ are given by at the end:

$$\mathbf{x}(t) = \sum_{n=1}^{N}\sum_{m(n)} \tilde{\mathbf{G}}_{m_1(n)}(s_1)e^{(m_1(n)s_1)t}, \; \dot{\mathbf{x}}(t) = \sum_{n=1}^{N}\sum_{m(n)} \tilde{\mathbf{G}}_{m_1(n)}(s_1)m_1(n)s_1e^{(m_1(n)s_1)t} \tag{A.3}$$

By substituting in the differential equation of the system Eq. (5.1), we have

$$\begin{aligned}\dot{\mathbf{x}}(t) - \mathbf{A}\mathbf{x}(t) - \mathbf{B}u(t) &= \sum_{n=1}^{N}\sum_{m(n)} (m_1(n)s_1\mathbf{I} - \mathbf{A})\tilde{\mathbf{G}}_{m_1(n)}(s_1)e^{(m_1(n)s_1)t} - \mathbf{B}e^{s_1t} \\ &= \mathbf{Q}\left(\sum_{n=1}^{N}\sum_{m(n)} \tilde{\mathbf{G}}_{m_1(n)}(s_1)e^{(m_1(n)s_1)t} \otimes \sum_{n=1}^{N}\sum_{m(n)} \tilde{\mathbf{G}}_{m_1(n)}(s_1)e^{(m_1(n)s_1)t}\right).\end{aligned} \tag{A.4}$$

By collecting the terms with e^{s_1t}, we result to

$$(s_1\mathbf{I} - \mathbf{A})\tilde{\mathbf{G}}_1(s_1) = \mathbf{B} \Rightarrow \tilde{\mathbf{G}}_1(s_1) = (s_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}. \tag{A.5}$$

We adjust in a similar way as in H_n the weighted $\mathbf{G}_1(s_1) = \frac{1!}{m_1(1)}\tilde{\mathbf{G}}_1(s_1) = (s_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$. Multiplication with the vector $\mathbf{C}$ from the left gives the 1st order GFRF consistent with the linear subsystem and can be simplified further using the resolvent notation.

$$H_1(s_1) = \mathbf{C}(s_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \mathbf{C}\,\underbrace{\mathbf{\Phi}(s_1)\overbrace{\mathbf{B}}^{\mathbf{R}_1}}_{\mathbf{G}_1(s_1)}. \tag{A.6}$$

Higher-order kernels can be derived at this level, e.g., $H_2, H_3, \ldots$. Still, these can be evaluated only on the diagonal of the hyper-plane that spans the domain of definition, e.g., $H_2(s_1, s_1)$, $H_3(s_1, s_1, s_1)$ which is not enough for achieving the identification goal as H_2 has a 2D domain support where a single harmonic input will always give information on the univariate diagonal (NFR method). Therefore, the next step is to excite with more complex inputs in terms of harmonics to identify the structure of the higher kernels.

- $R = 2$ **- 2nd order GFRF** $H_2(s_1, s_2)$**:** With input $u(t) = e^{s_1t} + e^{s_2t}$ the state solution is:

$$\mathbf{x}(t) = \sum_{n=1}^{N}\sum_{m(n)} \tilde{\mathbf{G}}_{m_1(n)m_2(n)}(s_1, s_2)e^{(m_1(n)s_1 + m_2(n)s_2)t}, \tag{A.7}$$

and the time derivative results to

$$\dot{\mathbf{x}}(t) = \sum_{n=1}^{N}\sum_{m(n)} (m_1(n)s_1 + m_2(n)s_2)\tilde{\mathbf{G}}_{m_1(n)m_2(n)}(s_1, s_2)e^{(m_1(n)s_1 + m_2(n)s_2)t}. \tag{A.8}$$

By substituting into the differential equation of the system (5.1), we obtain

$$\begin{aligned}
&\dot{\mathbf{x}}(t) - \mathbf{A}\mathbf{x}(t) = \\
&= \sum_{n=1}^{N} \sum_{m(n)} ((m_1(n)s_1 + m_2(n)s_2)\mathbf{I} - \mathbf{A})\tilde{\mathbf{G}}_{m_1(n)m_2(n)}(s_1, s_2)e^{(m_1(n)s_1+m_2(n)s_2)t} \\
&= \mathbf{Q}\left(\sum_{n=1}^{N} \sum_{m(n)} \tilde{\mathbf{G}}_{m_1(n)m_2(n)}(s_1, s_2)e^{(m_1(n)s_1+m_2(n)s_2)t} \otimes \ldots\right. \\
&\left.\ldots \otimes \sum_{n=1}^{N} \sum_{m(n)} \tilde{\mathbf{G}}_{m_1(n)m_2(n)}(s_1, s_2)e^{(m_1(n)s_1+m_2(n)s_2)t}\right) + \mathbf{B}(e^{s_1 t} + e^{s_2 t}).
\end{aligned} \tag{A.9}$$

By collecting the terms $e^{s_1 t + s_2 t}$ with $(n = 2,\ m_1(2) = 1,\ m_2(2) = 1)$, we result to

$$\begin{aligned}
&((s_1 + s_2)\mathbf{I} - \mathbf{A}))\,\tilde{\mathbf{G}}_{11}(s_1, s_2)e^{(s_1+s_2)t} = \\
&= \mathbf{Q}\left[\tilde{\mathbf{G}}_{10}(s_1)e^{s_1 t} \otimes \tilde{\mathbf{G}}_{01}(s_2)e^{s_2 t} + \tilde{\mathbf{G}}_{01}(s_2)e^{s_2 t} \otimes \tilde{\mathbf{G}}_{10}(s_1)e^{s_1 t}\right] \Rightarrow \\
&((s_1 + s_2)\mathbf{I} - \mathbf{A}))\,\tilde{\mathbf{G}}_{11}(s_1, s_2) = \mathbf{Q}\left[\tilde{\mathbf{G}}_{10}(s_1) \otimes \tilde{\mathbf{G}}_{01}(s_2) + \tilde{\mathbf{G}}_{01}(s_2) \otimes \tilde{\mathbf{G}}_{10}(s_1)\right] \Rightarrow \\
&\tilde{\mathbf{G}}_{11}(s_1, s_2) = ((s_1 + s_2)\mathbf{I} - \mathbf{A})^{-1} \cdot \\
&\cdot \mathbf{Q}\left[(s_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \otimes (s_2\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + (s_2\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \otimes (s_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\right]
\end{aligned} \tag{A.10}$$

Finally, by adjusting the weighted $\tilde{\mathbf{G}}_{11}(s_1, s_2) = \frac{2!}{1!1!}\mathbf{G}_2(s_1, s_2) = 2\mathbf{G}_2(s_1, s_2)$, and multiplying from the left with $\mathbf{C}$, we can define the 2nd order GFRF after using the resolvent notation as

$$\begin{aligned}
H_2(s_1, s_2) &= \frac{1}{2}\mathbf{C}\mathbf{\Phi}(s_1 + s_2)\mathbf{Q}\underbrace{[\mathbf{\Phi}(s_1)\mathbf{B} \otimes \mathbf{\Phi}(s_2)\mathbf{B} + \mathbf{\Phi}(s_2)\mathbf{B} \otimes \mathbf{\Phi}(s_1)\mathbf{B}]}_{\mathbf{R}_2(s_1,s_2)} \\
&= \mathbf{C}\underbrace{\frac{1}{2}\mathbf{\Phi}(s_1 + s_2)\mathbf{Q}\left[\mathbf{G}_1(s_1) \otimes \mathbf{G}_1(s_2) + \mathbf{G}_1(s_2) \otimes \mathbf{G}_1(s_1)\right]}_{\mathbf{G}_2(s_1,s_2)}
\end{aligned} \tag{A.11}$$

- $\underline{R = 3}$ **<u>- 3rd order GFRF</u>** $\underline{H_3(s_1, s_2, s_3)}$**<u>:</u>** With input $u(t) = e^{s_1 t} + e^{s_2 t} + e^{s_3 t}$, and similar arguments, we can derive

$$\begin{aligned}
H_3(s_1, s_2, s_3) &= \mathbf{C}\underbrace{\frac{1}{6}\mathbf{\Phi}(s_1 + s_2 + s_3)\mathbf{Q}\mathbf{R}_3(s_1, s_2, s_3)}_{\mathbf{G}_3(s_1,s_2,s_3)},\ \text{with} \\
\mathbf{R}_3(s_1, s_2, s_3) = &\mathbf{G}_1(s_1) \otimes \mathbf{G}_2(s_2, s_3) + \mathbf{G}_2(s_2, s_3) \otimes \mathbf{G}_1(s_1) + \\
&\mathbf{G}_1(s_2) \otimes \mathbf{G}_2(s_1, s_3) + \mathbf{G}_2(s_1, s_3) \otimes \mathbf{G}_1(s_2) + \\
&\mathbf{G}_1(s_3) \otimes \mathbf{G}_2(s_1, s_2) + \mathbf{G}_2(s_1, s_2) \otimes \mathbf{G}_1(s_3).
\end{aligned} \tag{A.12}$$

APPENDIX B

APPENDIXB

The 1st Volterra kernel for $m=1,\ u(t)=e^{s_1t},\ \mathbf{x}(t)=\sum_{k_1=0}^{\infty}\mathbf{G}_{k_1}(k_1s_1)e^{k_1s_1t}$ and substituting in the system, we can derive

$$\begin{aligned}\sum_{k_1=0}^{\infty}(k_1s_1\mathbf{I}-\mathbf{A})\mathbf{G}_{k_1}(k_1s_1)e^{k_1s_1t}=\mathbf{B}e^{s_1t}+\mathbf{Q}(\sum_{k_1=0}^{\infty}\mathbf{G}_{k_1}(k_1s_1)e^{k_1s_1t}\otimes\sum_{k_1=0}^{\infty}\mathbf{G}_{k_1}(k_1s_1)e^{k_1s_1t})+\\ +\mathbf{N}\sum_{k_1=0}^{\infty}\mathbf{G}_{k_1}(k_1s_1)e^{k_1s_1t}e^{s_1t},\end{aligned}\tag{B.1}$$

as we want to extract $\mathbf{G}_1$, we set $k_1=1$ and collect the same exponential powers

$$(s_1\mathbf{I}-\mathbf{A})\mathbf{G}_1(s_1)=\mathbf{B}\Rightarrow\mathbf{G}_1(s_1)=(s_1\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}.\tag{B.2}$$

The 2nd Volterra kernel for $m=2,\ u(t)=e^{s_1t}+e^{s_2t},$

$$\mathbf{x}(t)=\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\mathbf{G}_{k_1+k_2}(k_1s_1,k_2s_2)e^{(k_1s_1+k_2s_2)t}$$

and substituting to the system, it remains

$$\begin{aligned}&\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}((k_1s_1+k_2s_2)\mathbf{I}-\mathbf{A})\mathbf{G}_{k_1+k_2}(k_1s_1,k_2s_2)e^{(k_1s_1+k_2s_2)t}=\mathbf{B}(e^{s_1t}+e^{s_2t})+\\ &+\mathbf{Q}(\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\mathbf{G}_{k_1+k_2}(k_1s_1,k_2s_2)e^{(k_1s_1+k_2s_2)t}\otimes\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\mathbf{G}_{k_1+k_2}(k_1s_1,k_2s_2)e^{(k_1s_1+k_2s_2)t})+\\ &+\mathbf{N}\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\mathbf{G}_{k_1+k_2}(k_1s_1,k_2s_2)e^{(k_1s_1+k_2s_2)t}(e^{s_1t}+e^{s_2t}),\end{aligned}\tag{B.3}$$

as we want to extract $\mathbf{G}_2(s_1,s_2)$, we collect the same exponential powers with (s_1+s_2).

$$\begin{aligned}&((s_1+s_2)\mathbf{I}-\mathbf{A})\mathbf{G}_2(s_1,s_2)=\mathbf{Q}(\mathbf{G}_1(s_1)\otimes\mathbf{G}_1(s_2)+\mathbf{G}_1(s_2)\otimes\mathbf{G}_1(s_1))+\mathbf{N}(\mathbf{G}_1(s_1)+\mathbf{G}_1(s_2))\Rightarrow\\ &\mathbf{G}_2(s_1,s_2)=((s_1+s_2)\mathbf{I}-\mathbf{A})^{-1}\mathbf{Q}(\mathbf{G}_1(s_1)\otimes\mathbf{G}_1(s_2)+\\ &+\mathbf{G}_1(s_2)\otimes\mathbf{G}_1(s_1))+((s_1+s_2)\mathbf{I}-\mathbf{A})^{-1}\mathbf{N}(\mathbf{G}_1(s_1)+\mathbf{G}_1(s_2))\end{aligned}\tag{B.4}$$

The 3rd Volterra kernel is for $m = 3,\ u(t) = e^{s_1 t} + e^{s_2 t} + e^{s_3 t}$,

$$\mathbf{x}(t) = \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty} \mathbf{G}_{k_1+k_2+k_3}(k_1s_1, k_2s_2, k_3s_3)e^{(k_1s_1+k_2s_2+k_3s_3)t}$$

and substituting to the system

$$\begin{aligned}
&\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}((k_1s_1+k_2s_2+k_3s_3)\mathbf{I}-\mathbf{A})\mathbf{G}_{k_1+k_2+k_3}(k_1s_1,k_2s_2,k_3s_3)e^{(k_1s_1+k_2s_2+k_3s_3)t} = \\
&= \mathbf{B}(e^{s_1t}+e^{s_2t}+e^{s_3t}) + \\
&+\mathbf{Q}(\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\mathbf{G}_{k_1+k_2+k_3}(k_1s_1,k_2s_2,k_3s_3)e^{(k_1s_1+k_2s_2+k_3s_3)t}\otimes \\
&\otimes\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\mathbf{G}_{k_1+k_2+k_3}(k_1s_1,k_2s_2,k_3s_3)e^{(k_1s_1+k_2s_2+k_3s_3)t})+ \\
&+\mathbf{N}\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\mathbf{G}_{k_1+k_2+k_3}(k_1s_1,k_2s_2,k_3s_3)e^{(k_1s_1+k_2s_2+k_3s_3)t}(e^{s_1t}+e^{s_2t}+e^{s_3t})
\end{aligned} \tag{B.5}$$

As we want to extract $\mathbf{G}_3(s_1, s_2, s_3)$, we collect the same exponential powers with $(s_1 + s_2 + s_3)$.

$$\begin{aligned}
((s_1+s_2+s_3)\mathbf{I}-\mathbf{A})\mathbf{G}_3(s_1,s_2,s_3) =&\mathbf{Q}\Big(\mathbf{G}_1(s_1)\otimes\mathbf{G}_2(s_2,s_3)+\mathbf{G}_2(s_2,s_3)\otimes\mathbf{G}_1(s_1)+ \\
&+\mathbf{G}_1(s_2)\otimes\mathbf{G}_2(s_1,s_3)+\mathbf{G}_2(s_1,s_3)\otimes\mathbf{G}_1(s_2)+ \\
&+\mathbf{G}_1(s_3)\otimes\mathbf{G}_2(s_1,s_2)+\mathbf{G}_2(s_1,s_2)\otimes\mathbf{G}_1(s_3)\Big)+ \\
&\mathbf{N}\Big(\mathbf{G}_2(s_1,s_2)+\mathbf{G}_2(s_1,s_3)+\mathbf{G}_2(s_2,s_3)\Big)
\end{aligned} \tag{B.6}$$

The 4th Volterra kernel is with $m = 4,\ u(t) = e^{s_1 t} + e^{s_2 t} + e^{s_3 t} + e^{s_4 t}$,

$$\mathbf{x}(t) = \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\sum_{k_4=0}^{\infty} \mathbf{G}_{k_1+k_2+k_3+k_4}(k_1s_1, k_2s_2, k_3s_3, k_4s_4)e^{(k_1s_1+k_2s_2+k_3s_3+k_4s_4)t}$$

and substituting to the system

$$\begin{aligned}
&\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\sum_{k_4=0}^{\infty}((k_1s_1+\cdots+k_4s_4)\mathbf{I}-\mathbf{A})\mathbf{G}_{k_1+\cdots+k_4}(k_1s_1,\ldots,k_4s_4)e^{(k_1s_1+\cdots+k_4s_4)t} = \\
&= \mathbf{B}(e^{s_1t}+e^{s_2t}+e^{s_3t}+e^{s_4t}) + \\
&+\mathbf{Q}(\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\sum_{k_4=0}^{\infty}\mathbf{G}_{k_1+k_2+k_3+k_4}(k_1s_1,k_2s_2,k_3s_3,k_4s_4)e^{(k_1s_1+k_2s_2+k_3s_3+k_4s_4)t}\otimes \\
&\otimes\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\sum_{k_4=0}^{\infty}\mathbf{G}_{k_1+k_2+k_3+k_4}(k_1s_1,k_2s_2,k_3s_3,k_4s_4)e^{(k_1s_1+k_2s_2+k_3s_3+k_4s_4)t})+ \\
&+\mathbf{N}\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\sum_{k_4=0}^{\infty}\mathbf{G}_{k_1+\cdots+k_4}(k_1s_1,\ldots,k_4s_4)e^{(k_1s_1+\cdots+k_4s_4)t}(e^{s_1t}+e^{s_2t}+e^{s_3t}+e^{s_4t})
\end{aligned} \tag{B.7}$$

As we want to extract $\mathbf{G}_4(s_1, s_2, s_3, s_4)$, we collect the same exponential powers with $(s_1 + s_2 + s_3 + s_4)$.

$$\begin{aligned}
((s_1 + s_2 + s_3 + s_4)\mathbf{I} - \mathbf{A})\mathbf{G}_4(s_1, s_2, s_3, s_4) = \mathbf{Q}\Big(&\mathbf{G}_1(s_1) \otimes \mathbf{G}_3(s_2, s_3, s_4) + \mathbf{G}_3(s_2, s_3, s_4) \otimes \mathbf{G}_1(s_1) \\
&+ \mathbf{G}_1(s_2) \otimes \mathbf{G}_3(s_3, s_4, s_1) + \mathbf{G}_3(s_3, s_4, s_1) \otimes \mathbf{G}_1(s_2) \\
&+ \mathbf{G}_1(s_3) \otimes \mathbf{G}_3(s_4, s_1, s_2) + \mathbf{G}_3(s_4, s_1, s_2) \otimes \mathbf{G}_1(s_3) \\
&+ \mathbf{G}_1(s_4) \otimes \mathbf{G}_3(s_1, s_2, s_3) + \mathbf{G}_3(s_1, s_2, s_3) \otimes \mathbf{G}_1(s_4) \\
&+ \mathbf{G}_2(s_1, s_2) \otimes \mathbf{G}_2(s_3, s_4) + \mathbf{G}_2(s_3, s_4) \otimes \mathbf{G}_2(s_1, s_2) \\
&+ \mathbf{G}_2(s_1, s_3) \otimes \mathbf{G}_2(s_2, s_4) + \mathbf{G}_2(s_2, s_4) \otimes \mathbf{G}_2(s_1, s_3) \\
&+ \mathbf{G}_2(s_1, s_4) \otimes \mathbf{G}_2(s_2, s_3) + \mathbf{G}_2(s_2, s_3) \otimes \mathbf{G}_2(s_1, s_4) \\
&+ \mathbf{G}_2(s_2, s_3) \otimes \mathbf{G}_2(s_1, s_4) + \mathbf{G}_2(s_1, s_4) \otimes \mathbf{G}_2(s_2, s_3)\Big) \\
&\mathbf{N}\big(\mathbf{G}_3(s_1, s_2, s_3) + \mathbf{G}_3(s_2, s_3, s_4) + \mathbf{G}_3(s_3, s_4, s_1) + \mathbf{G}_3(s_4, s_1, s_2)\big)
\end{aligned} \tag{B.8}$$

The $\mathbf{R}_\mathbf{Q}(s_1, s_2, \ldots, s_m)$ and $\mathbf{R}_\mathbf{N}(s_1, s_2, \ldots, s_m)$ are defined up to $m = 4$. Although tedious, the construction pattern is revealed by considering all the combinations. Further, the weighted symmetric kernels are $\mathbf{H}_m(s_1, \ldots, s_m) = \frac{1}{m!}\mathbf{G}_m(s_1, \ldots, s_m)$.

APPENDIX C

APPENDIXC

Dynamical systems that form an equivalent modulo can be aligned using a similarity transform $\mathbf{\Psi} \in \mathbb{R}^{n \times n}$ as:

$$\mathbf{\Phi}_1 = \left[\begin{array}{cccc} (\mathbf{C}_1\mathbf{A}_1)^T & (\mathbf{C}_1\mathbf{A}_1^2)^T & \cdots & (\mathbf{C}_1\mathbf{A}_1^n)^T \end{array}\right], \; \mathbf{\Phi}_2 = \left[\begin{array}{cccc} (\mathbf{C}_2\mathbf{A}_2)^T & (\mathbf{C}_2\mathbf{A}_2^2)^T & \cdots & (\mathbf{C}_2\mathbf{A}_2^n)^T \end{array}\right]^T,$$
$$\mathbf{\Psi} = \mathbf{\Phi}_2^{-1}\mathbf{\Phi}_1, \text{ and for the operators of the quadratic system holds:}$$
$$(\mathbf{A}_2,\ \mathbf{Q}_2,\ \mathbf{B}_2,\ \mathbf{C}_2) = (\mathbf{\Psi}^{-1}\mathbf{A}_1\mathbf{\Psi},\ \mathbf{\Psi}^{-1}\mathbf{Q}_1(\mathbf{\Psi} \otimes \mathbf{\Psi}),\ \mathbf{\Psi}^{-1}\mathbf{B}_1,\ \mathbf{C}_1\mathbf{\Psi}). \tag{C.1}$$

BIBLIOGRAPHY

[1] M. I. Ahmad, U. Baur, and P. Benner, *Implicit Volterra series interpolation for model reduction of bilinear systems*, J. Comput. Appl. Math., 316 (2017), pp. 15–28, https://doi.org/10.1016/j.cam.2016.09.048. 69

[2] N. I. Akhiezer, *Elements of the theory of elliptic functions*, vol. 79 of Translations of Mathematical Monographs, American Mathematical Society, Providence, RI, 1990. Translated from the second Russian edition by H. H. McFaden. 59, 60

[3] S. A. Al-Baiyat, *Model reduction of bilinear systems described by input-output difference equation*, International Journal of Systems Science, 35 (2004), pp. 503–510, https://doi.org/10.1080/00207720410001734237. 102, 119

[4] A. Antoulas, *The Loewner framework and transfer functions of singular/rectangular systems*, Applied Mathematics Letters, 54 (2016), pp. 36 – 47, https://doi.org/https://doi.org/10.1016/j.aml.2015.10.011. 29

[5] A. Antoulas, I. Gosea, and M. Heinkenschloss, *Reduction of systems with polynomial nonlinearities in the Loewner framework*, in Book of Abstracts of XXI Householder Symposium on Numerical Linear Algebra, Selva di Fasano, Italy, June 14–19, 2020, https://users.ba.cnr.it//iac/irmanm21/HHXXI/book_of_abstracts_HHXXI0.pdf. 29, 84

[6] A. Antoulas, A. Ionita, and S. Lefteriu, *On two-variable rational interpolation*, Linear Algebra and its Applications, 436 (2012), pp. 2889 – 2915, https://doi.org/https://doi.org/10.1016/j.laa.2011.07.017. Special Issue dedicated to Danny Sorensen's 65th birthday. 29

[7] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, Society for Industrial and Applied Mathematics, 2005, https://doi.org/10.1137/1.9780898718713. 1, 8, 33, 41, 155

[8] A. C. Antoulas, C. A. Beattie, and S. Gugercin, *Interpolatory Methods for Model Reduction*, SIAM, Philadelphia, 2020. 1, 2, 12, 17, 29, 83, 155, 157, 158

[9] A. C. Antoulas, I. V. Gosea, and M. Heinkenschloss, *On the Loewner framework for model reduction of Burgers' equation*, in Active Flow and Combustion Control, R. King, ed., Springer, Cham, Switzerland, 2019, pp. 255–270. 3, 141, 145

[10] A. C. Antoulas, I. V. Gosea, and M. Heinkenschloss, *On the Loewner framework for model reduction of Burgers' equation*, in Active Flow and Combustion Control 2018, R. King, ed., Cham, 2019, Springer International Publishing, pp. 255–270. 29

[11] A. C. Antoulas, I. V. Gosea, and M. Heinkenschloss, *Data-driven model reduction for a class of semi-explicit daes using the Loewner framework*, in Progress in Differential-Algebraic Equations II, T. Reis, S. Grundel, and S. Schöps, eds., Cham, 2020, Springer International Publishing, pp. 185–210. 29

[12] A. C. Antoulas, I. V. Gosea, and A. C. Ionita, *Model reduction of bilinear systems in the Loewner framework*, SIAM Journal on Scientific Computing, 38 (2016), pp. B889–B916, https://doi.org/10.1137/15M1041432, https://doi.org/10.1137/15M1041432. 2, 29, 68, 69, 80, 82, 83, 84, 85, 86, 87, 90, 92, 93, 97, 98

[13] A. C. Antoulas, S. Lefteriu, and A. C. Ionita, *A Tutorial Introduction to the Loewner Framework for Model Reduction*, Society for Industrial and Applied Mathematics for Computational Science & Engineering, 2017, ch. 8, pp. 335–376, https://doi.org/10.1137/1.9781611974829.ch8. 16, 29, 75, 155, 157, 158

[14] A. C. Antoulas, B. Zhu, Q. Zhang, B. York, B. W. O'Malley, and C. C. Dacso, *A novel mathematical method for disclosing oscillations in gene transcription: A comparative study*, PLOS ONE, 13 (2018), pp. 1–20, https://doi.org/10.1371/journal.pone.0198503. 29

[15] Z. Bai and D. Skoogh, *A projection method for model reduction of bilinear dynamical systems*, Linear Algebra and its Applications, 415 (2006), pp. 406–425, https://doi.org/https://doi.org/10.1016/j.laa.2005.04.032. Special Issue on Order Reduction of Large-Scale Systems. 69

[16] C. Baker, K. Gallivan, and P. V. Dooren, *Low-rank incremental methods for computing dominant singular subspaces*, Linear Algebra and its Applications, 436 (2012), pp. 2866 – 2888, https://doi.org/https://doi.org/10.1016/j.laa.2011.07.018. Special Issue dedicated to Danny Sorensen's 65th birthday. 9

[17] U. Baur, P. Benner, and L. Feng, *Model order reduction for linear and nonlinear systems: a system-theoretic perspective*, Archives of Computational Methods in Engineering, 21 (2014), pp. 331–358. 1

[18] C. Beattie and S. Gugercin, *Realization-independent H2-approximation*, in 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Dec 2012, pp. 4953–4958, https://doi.org/10.1109/CDC.2012.6426344. 28, 56, 57

[19] C. Beattie and S. Gugercin, *Model Reduction by Rational Interpolation*, Society for Industrial and Applied Mathematics for Computational Science & Engineer-

ing, 2017, ch. 7, pp. 297–334, https://doi.org/10.1137/1.9781611974829.ch7. 56

[20] C. Beattie, S. Gugercin, and V. Mehrmann, *Model reduction for systems with inhomogeneous initial conditions*, Systems & Control Letters, 99 (2017), pp. 99–106, https://doi.org/https://doi.org/10.1016/j.sysconle.2016.11.007. 65

[21] B. Beckermann and A. Townsend, *On the singular values of matrices with displacement structure*, SIAM Journal on Matrix Analysis and Applications, 38 (2017), pp. 1227–1248, https://doi.org/10.1137/16M1096426. 47

[22] P. Benner and T. Breiten, *Interpolation-based $h_ 2$-model reduction of bilinear control systems*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 859–885, https://doi.org/10.1137/110836742. 69

[23] P. Benner, T. Breiten, and T. Damm, *Generalised tangential interpolation for model reduction of discrete-time mimo bilinear systems*, International Journal of Control, 84 (2011), pp. 1398–1407, https://doi.org/10.1080/00207179.2011.601761. 69, 115

[24] P. Benner and L. Feng, *Model order reduction based on moment-matching*, in Model Order Reduction: Volume 1: System-and Data-Driven Methods and Algorithms, De Gruyter, 2021, pp. 57–96. 1

[25] P. Benner and P. Goyal, *Interpolation-based model order reduction for polynomial systems*, SIAM Journal on Scientific Computing, 43 (2021), pp. A84–A108, https://doi.org/10.1137/19M1259171. 3, 26

[26] P. Benner and P. Goyal, *Interpolation-based model order reduction for polynomial systems*, SIAM Journal on Scientific Computing, 43 (2021), pp. A84–A108, https://doi.org/10.1137/19M1259171. 29, 84

[27] P. Benner, P. Goyal, J. Heiland, and I. Pontes Duff, *Operator inference and physics-informed learning of low-dimensional models for incompressible flows*, e-prints 2010.06701, arXiv, 2020, http://arxiv.org/abs/2010.06701. math.DS. 146

[28] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox, *Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms*, Computer Methods in Applied Mechanics and Engineering, 372 (2020), p. 113433, https://doi.org/https://doi.org/10.1016/j.cma.2020.113433. 2, 149

[29] P. Benner, P. Goyal, and P. Van Dooren, *Identification of port-hamiltonian systems from frequency response data*, Systems & Control Letters, 143 (2020), p. 104741, https://doi.org/https://doi.org/10.1016/j.sysconle.2020.104741. 29

[30] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, eds., *Model Order Reduction Volume 2: Snapshot-Based Methods and Algorithms*, De Gruyter, Berlin, Boston, 2020, https://doi.org/doi:10.1515/9783110671490. 1

[31] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, eds., *Model Order Reduction Volume 3: Applications*, De Gruyter, Berlin, Boston, 2020, https://doi.org/doi:10.1515/9783110499001. 1

[32] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, eds., *2 Balancing-related model reduction methods*, De Gruyter, Berlin, Boston, 2021, pp. 15–56, https://doi.org/doi:10.1515/9783110498967-002. 1

[33] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, eds., *Model Order Reduction Volume 1: System- and Data-Driven Methods and Algorithms*, De Gruyter, Berlin, Boston, 2021, https://doi.org/doi:10.1515/9783110498967. 1

[34] M. Berljafa and S. Güttel, *A rational Krylov toolbox for Matlab*, 2014. 60

[35] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, Wiley, 2013. 175

[36] S. Boyd and L. S. Chua, *Fading memory and the problem of approximating nonlinear operators with Volterra series*, IEEE Transactions on Circuits and Systems, 30 (1985), pp. 1150–1161. 136

[37] S. Boyd, Y. shing Tang, and L. O. Chua, *Measuring Volterra kernels*, IEEE Transactions on Circuits and Systems, (1983). 20, 70, 74, 123, 129, 136, 139

[38] T. Breiten, *Interpolatory Methods for Model Reduction of Large-Scale Dynamical Systems*, Dissertation, Department of Mathematics, Otto-von-Guericke University, Magdeburg, Germany, 2013. 24, 69, 145, 147, 148

[39] P. D. Brubeck, Y. Nakatsukasa, and L. N. Trefethen, *Vandermonde with Arnoldi*, 2019, https://arxiv.org/abs/1911.09988. 20

[40] S. Brunton and J. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2022, https://books.google.de/books?id=rxNkEAAAQBAJ. 1, 2

[41] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proceedings of the National Academy of Sciences, 113 (2016), pp. 3932–3937, https://doi.org/10.1073/pnas.1517384113. 2

[42] B. C., D. G., and K. G., *On the mathematical models of bilinear systems.*, Ricerche di Automatica, 2 (1971), pp. 11–26. 73

[43] T. Carleman, *Application de la théories des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires*, Acta. Math., 59 (1932), pp. 63–87. 21

[44] H. Chen and J. Maciejowski, *Subspace identification method for combined deterministic-stochastic bilinear systems*, IFAC Proceedings Volumes, 33 (2000), pp. 229–234, https://doi.org/https://doi.org/10.1016/S1474-6670(17)39755-0. 12th IFAC Symposium on System Identification (SYSID 2000), Santa Barbara, CA, USA, 21-23 June 2000. 2, 102

[45] S. Chen and S. A. Billings, *Representations of non-linear systems: the narmax model*, International Journal of Control, 49 (1989), pp. 1013–1032, https://doi.org/10.1080/00207178908559683. 2, 118, 119

[46] Y. Chen, *Model reduction for nonlinear systems*, master's thesis, M.I.T, Boston, USA, 1999. 101, 147, 148

[47] R. Curtain and K. Morris, *Transfer functions of distributed parameter systems: A tutorial*, Automatica, 45 (2009), pp. 1101 – 1116, https://doi.org/https://doi.org/10.1016/j.automatica.2009.01.008. 55

[48] A. N. Diaz and M. Heinkenschloss, *Towards data-driven model reduction of the Navier-Stokes equations using the Loewner framework*, in Active Flow and Combustion Control 2021, R. King and D. Peitsch, eds., Cham, 2022, Springer International Publishing, pp. 225–239. 29

[49] H. Dorissen, *A method for bilinear system identification*, IFAC Proceedings Volumes, 23 (1990), pp. 143–148, https://doi.org/https://doi.org/10.1016/S1474-6670(17)51998-9. 11th IFAC World Congress on Automatic Control, Tallinn, 1990 - Volume 2, Tallinn, Finland. 102, 107, 108

[50] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Relative-error CUR matrix decompositions*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), pp. 844–881, https://doi.org/10.1137/07070471X. 9

[51] Z. Drmač and B. Peherstorfer, *Learning Low-Dimensional Dynamical-System Models from Noisy Frequency-Response Data with Loewner Rational Interpolation*, Springer International Publishing, Cham, 2022, pp. 39–57, https://doi.org/10.1007/978-3-030-95157-3_3. 29

[52] Z. Drmač, S. Gugercin, and C. Beattie, *Vector fitting for matrix-valued rational approximation*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2346–A2379, https://doi.org/10.1137/15M1010774. 16, 28

[53] A. Dunoyer, L. Balmer, K. J. Burnham, and D. J. G. James, *On the discretization of single-input single-output bilinear systems*, International Journal of Control, 68 (1997), pp. 361–372, https://doi.org/10.1080/002071797223668. 105

[54] W. Favoreel, B. De Moor, and P. Van Overschee, *Subspace identification of bilinear systems subject to white inputs*, IEEE Transactions on Automatic Control, 44 (1999), pp. 1157–1165, https://doi.org/10.1109/9.769370. 2, 101, 102, 103

[55] G. Flagg and S. Gugercin, *Multipoint Volterra series interpolation and H_2 optimal model reduction of bilinear systems*, SIAM Journal on Matrix Analysis and Applications, 36 (2015), pp. 549–579, https://doi.org/10.1137/130947830. 87

[56] G. Flagg and S. Gugercin, *Multipoint Volterra series interpolation and $\mathcal{H}_2$ optimal model reduction of bilinear systems*, SIAM Journal on Matrix Analysis and Applications, 36 (2015), pp. 549–579, https://doi.org/10.1137/130947830. 69

[57] K. Gallivan, A. Vandendorpe, and P. V. Dooren, *On the generality of multipoint Padé approximations*, IFAC Proceedings Volumes, 35 (2002), pp. 331 – 336, https://doi.org/https://doi.org/10.3182/20020721-6-ES-1901.00543. 15th IFAC World Congress. 13

[58] K. Gallivan, A. Vandendorpe, and P. V. Dooren, *Sylvester equations and projection-based model reduction*, Journal of Computational and Applied Mathematics, 162 (2004), pp. 213 – 229, https://doi.org/https://doi.org/10.1016/j.cam.2003.08.026. Proceedings of the International Conference on Linear Algebra and Arithmetic 2001. 13

[59] K. Gallivan, A. Vandendorpe, and P. Van Dooren, *Model reduction of MIMO systems via tangential interpolation*, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 328–349, https://doi.org/10.1137/S0895479803423925. 16

[60] S. Gao, *Factoring multivariate polynomials via partial differential equations*, Math. Comput., 72 (2003), pp. 801–822. 98

[61] S. Goreinov and E. Tyrtyshnikov, *Quasioptimality of Skeleton approximation of a matrix in the Chebyshev norm*, In: Doklady Mathematics ed: Springer, 83 (2011), pp. 374–375, https://doi.org/https://doi.org/10.1134/S1064562411030355. 9, 44

[62] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra and its Applications, 261 (1997), pp. 1 – 21, https://doi.org/https://doi.org/10.1016/S0024-3795(96)00301-1. 9, 44

[63] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *How to Find a Good Submatrix*, World Scientific, 2010, pp. 247–256, https://doi.org/10.1142/9789812836021_0015. 9

[64] I. V. Gosea, *Exact and inexact lifting transformations of nonlinear dynamical systems: Transfer functions, equivalence, and complexity reduction*, Applied Sciences, 12 (2022), https://doi.org/10.3390/app12052333. 18, 24

[65] I. V. Gosea and A. C. Antoulas, *Model reduction of linear and nonlinear systems in the Loewner framework: A summary*, in 2015 European Control Conference (ECC), 2015, pp. 345–349, https://doi.org/10.1109/ECC.2015.7330568. 3, 29

[66] I. V. Gosea and A. C. Antoulas, *Stability preserving post-processing methods applied in the Loewner framework*, in 2016 IEEE 20th Workshop on Signal and Power Integrity (SPI), 2016, pp. 1–4, https://doi.org/10.1109/SaPIW.2016.7496283. 29

[67] I. V. Gosea and A. C. Antoulas, *Approximation of a damped Euler-Bernoulli beam model in the Loewner framework*, 2017, https://arxiv.org/abs/1712.06031. 55

[68] I. V. Gosea and A. C. Antoulas, *Data-driven model order reduction of quadratic-bilinear systems*, Numerical Linear Algebra with Applications, 25 (2018), p. e2200, https://doi.org/10.1002/nla.2200. 3, 84, 145

[69] I. V. Gosea and A. C. Antoulas, *Data-driven model order reduction of quadratic-bilinear systems*, Numerical Linear Algebra with Applications, 25 (2018), https://doi.org/10.1002/nla.2200. e2200. 29

[70] I. V. Gosea and A. C. Antoulas, *The one-sided Loewner framework and connections to other model reduction methods based on interpolation*, IFAC-PapersOnLine, 55 (2022), pp. 377–382, https://doi.org/https://doi.org/10.1016/j.ifacol.2022.11.082. 25th International Symposium on Mathematical Theory of Networks and Systems MTNS 2022. 29

[71] I. V. Gosea, I. P. Duff, P. Benner, and A. C. Antoulas, *Model order reduction of bilinear time-delay systems*, in 2019 18th European Control Conference (ECC), 2019, pp. 2289–2294, https://doi.org/10.23919/ECC.2019.8796085. 29

[72] I. V. Gosea, D. S. Karachalios, and A. C. Antoulas, *Learning reduced order models of quadratic dynamical systems from input-output data*, in 2021 European Control Conference (ECC), 2021, pp. 1426–1431, https://doi.org/10.23919/ECC54610.2021.9654993. iv, 29, 122

[73] I. V. Gosea, D. S. Karachalios, and A. C. Antoulas, *On computing reduced order bilinear models from time-domain data*, PAMM, 21 (2021), p. e202100254, https://doi.org/https://doi.org/10.1002/pamm.202100254. iii, 29, 103

[74] I. V. Gosea, M. Petreczky, and A. C. Antoulas, *Data-driven model order reduction of linear switched systems in the Loewner framework*, SIAM Journal on Scientific Computing, 40 (2018), pp. B572–B610, https://doi.org/10.1137/17M1120233. 2

[75] I. V. Gosea, M. Petreczky, and A. C. Antoulas, *Data-driven model order reduction of linear switched systems in the Loewner framework*, SIAM Journal on Scientific Computing, 40 (2018), pp. B572–B610, https://doi.org/10.1137/17M1120233. 29

[76] I. V. Gosea, M. Petreczky, and A. C. Antoulas, *Reduced-order modeling of lpv systems in the Loewner framework*, in 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 3299–3305, https://doi.org/10.1109/CDC45484.2021.9683742. 2, 29

[77] I. V. Gosea, C. Poussot-Vassal, and A. C. Antoulas, *On enforcing stability for data-driven reduced-order models*, in 2021 29th Mediterranean Conference on Control and Automation (MED), 2021, pp. 487–493, https://doi.org/10.1109/MED51440.2021.9480216. 29

[78] I. V. Gosea, C. Poussot-Vassal, and A. C. Antoulas, *On Loewner data-driven control for infinite-dimensional systems*, in 2021 European Control Conference (ECC), 2021, pp. 93–99, https://doi.org/10.23919/ECC54610.2021.9655097. 29

[79] I. V. Gosea, C. Poussot-Vassal, and A. C. Antoulas, *Chapter 15 - data-driven modeling and control of large-scale dynamical systems in the Loewner framework: Methodology and applications*, in Numerical Control: Part A, E. Trélat and E. Zuazua, eds., vol. 23 of Handbook of Numerical Analysis, Elsevier, 2022, pp. 499–530, https://doi.org/https://doi.org/10.1016/bs.hna.2021.12.015. 29

[80] I. V. Gosea, Q. Zhang, and A. C. Antoulas, *Preserving the DAE structure in the Loewner model reduction and identification framework*, Adv. Comput. Math., 46 (2020), https://doi.org/https://doi.org/10.1007/s10444-020-09752-8. 29

[81] I. V. Gosea, L. A. Živković, D. S. Karachalios, T. Vidaković-Koch, and A. C. Antoulas, *A data-driven nonlinear frequency response approach based on the Loewner framework: preliminary analysis*, IFAC-PapersOnLine, 56 (2023), pp. 234–239, https://doi.org/https://doi.org/10.1016/j.ifacol.2023.02.040. 12th IFAC Symposium on Nonlinear Control Systems NOLCOS 2022. 29

[82] P. Goyal and P. Benner, *Discovery of nonlinear dynamical systems using a Runge–Kutta inspired dictionary-based sparse regression approach*, Proceedings of the Royal Society A, 478 (2022), p. 20210883, https://doi.org/10.1098/rspa.2021.0883. 1

[83] P. Goyal, I. Pontes Duff, and P. Benner, *Guaranteed stable quadratic models and their applications in SINDy and operator inference*, arXiv preprint, (2023), https://doi.org/10.48550/arXiv.2308.13819. 1

[84] P. Goyal, I. Pontes Duff, and P. Benner, *Inference of continuous linear systems from data with guaranteed stability*, arXiv preprint, (2023), https://doi.org/10.48550/arXiv.2301.10060. 2

[85] E. Grimme, K. A. Gallivan, and P. V. Dooren, *On some recent developments in projection-based model reduction*, in in ENUMATH 97 (Heidelberg), World Sci. Publishing, River Edge, NJ, 1998, pp. 98–113, https://api.semanticscholar.org/CorpusID:109243519. 13

[86] C. Gu, *A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30 (2011), pp. 1307–1320. 18, 24, 147

[87] S. Gugercin, A. C. Antoulas, and C. Beattie, *H2 model reduction for large-scale linear dynamical systems*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), pp. 609–638, https://doi.org/10.1137/060666123. 56

[88] B. Gustavsen and A. Semlyen, *Rational approximation of frequency domain responses by vector fitting*, IEEE Transactions on Power Delivery, 14 (1999), pp. 1052–1061, https://doi.org/10.1109/61.772353. 2, 28

[89] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Review, 53 (2011), pp. 217–288, https://doi.org/10.1137/090771806. 28

[90] M. Heinkenschloss, T. Reis, and A. Antoulas, *Balanced truncation model reduction for systems with inhomogeneous initial conditions*, Automatica, 47 (2011), pp. 559–564, https://doi.org/https://doi.org/10.1016/j.automatica.2010.12.002. 65

[91] S. M. Hirsh, K. D. Harris, J. N. Kutz, and B. W. Brunton, *Centering data improves the dynamic mode decomposition*, SIAM Journal on Applied Dynamical Systems, 19 (2020), pp. 1920–1955, https://doi.org/10.1137/19M1289881. 2

[92] B. L. Ho and R. E. Kalman, *Editorial: Effective construction of linear state-variable models from input/output functions*, at - Automatisierungstechnik, 14 (1966), pp. 545–548, https://doi.org/doi:10.1524/auto.1966.14.112.545. 2, 101, 111

[93] A. C. Ionita and A. C. Antoulas, *Data-driven parametrized model reduction in the Loewner framework*, SIAM Journal on Scientific Computing, 36 (2014), pp. A984–A1007, https://doi.org/10.1137/130914619. 29

[94] R. Ionutiu, J. Rommes, and A. C. Antoulas, *Passivity-preserving model reduction using dominant spectral-zero interpolation*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 27 (2008), pp. 2250–2263, https://doi.org/10.1109/TCAD.2008.2006160. 29

[95] A. Isidori, *Direct construction of minimal bilinear realizations from nonlinear input-output maps*, IEEE Transactions on Automatic Control, 18 (1973), pp. 626–631, https://doi.org/10.1109/TAC.1973.1100424. 2, 69, 94, 101, 108, 112, 113, 117

[96] A. Isidori, *Nonlinear Control Systems*, vol. 1, Springer, 1995. 17

[97] J.-N. Juang, *Continuous-time bilinear system identification*, Nonlinear Dynamics, 39 (2005), p. 79–94, https://doi.org/https://doi.org/10.1007/s11071-005-1915-z. 79

[98] J.-N. Juang, *Continuous-time bilinear system identification*, Nonlinear Dynamics, 39 (2005), p. 79–94, https://doi.org/https://doi.org/10.1007/s11071-005-1915-z. 103

[99] J.-N. Juang and R. S. Pappa, *An eigensystem realization algorithm for modal parameter identification and model reduction*, Journal of Guidance, Control, and Dynamics, 8 (1985), pp. 620–627, https://doi.org/10.2514/3.20031. 2, 101

[100] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas, *Data driven approximation methods applied to non-rational functions*, PAMM, 18 (2018), p. e201800368, https://doi.org/https://doi.org/10.1002/pamm.201800368. iii, 29, 53

[101] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas, *A bilinear identification-modeling framework from time-domain data*, PAMM, 19 (2019), p. e201900246, https://doi.org/https://doi.org/10.1002/pamm.201900246. iii, 29, 72, 78

[102] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas, *6 The Loewner framework for system identification and reduction*, De Gruyter, 2021, pp. 181–228, https://doi.org/doi:10.1515/9783110498967-006, https://doi.org/10.1515/9783110498967-006. iii, 75, 155, 157, 158

[103] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas, *The Loewner framework for nonlinear identification and reduction of Hammerstein cascaded dynamical systems*, PAMM, 20 (2021), p. e202000337, https://doi.org/https://doi.org/10.1002/pamm.202000337. iv, 155

[104] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas, *On Bilinear Time-Domain Identification and Reduction in the Loewner Framework*, Springer International Publishing, Cham, 2021, pp. 3–30, https://doi.org/10.1007/978-3-030-72983-7_1. iii, 29, 85, 99, 122, 123, 129, 136, 137, 139, 148

[105] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas, *A framework for fitting quadratic-bilinear systems with applications to models of electrical circuits*, IFAC-PapersOnLine, 55 (2022), pp. 7–12, https://doi.org/https://doi.org/10.1016/j.ifacol.2022.09.064. 10th Vienna International Conference on Mathematical Modelling MATHMOD 2022. iv, 29, 122, 126

[106] D. S. Karachalios, I. V. Gosea, L. Gkimisis, and A. C. Antoulas, *Data-driven quadratic modeling in the Loewner framework*, 2024, https://doi.org/10.48550/ARXIV.2211.10635, https://arxiv.org/abs/2211.10635. Applied Dynamical Systems. iv, 29

[107] D. S. Karachalios, I. V. Gosea, K. Kour, and A. C. Antoulas, *Bilinear realization from input-output data with neural networks*, 2022, https://doi.org/10.48550/ARXIV.2208.10124, https://arxiv.org/abs/2208.10124. Submitted in the SCEE proceedings on 31.07.2022, status: In Review. iii

[108] D. S. Karachalios, I. V. Gosea, Q. Zhang, and A. C. Antoulas, *Case study: Approximations of the Bessel function*, tech. report, arXiv preprint avaialable at https://arxiv.org/abs/1801.03390, 2017, https://arxiv.org/abs/1801.03390. 53

[109] P. Kergus and I. V. Gosea, *Data-driven approximation and reduction from noisy data in matrix pencils frameworks*, IFAC-PapersOnLine, 55 (2022), pp. 371–376, https://doi.org/https://doi.org/10.1016/j.ifacol.2022.11.081. 25th International Symposium on Mathematical Theory of Networks and Systems MTNS 2022. 29

[110] P. Kergus, C. Poussot-Vassal, F. Demourant, and S. Formentin, *Frequency-domain data-driven control design in the Loewner framework*, IFAC-PapersOnLine, 50 (2017), pp. 2095–2100, https://doi.org/https://doi.org/10.1016/j.ifacol.2017.08.531. 20th IFAC World Congress. 29

[111] P. Khodabakhshi and K. E. Willcox, *Non-intrusive data-driven model reduction for differential–algebraic equations derived from lifting transformations*, Computer Methods in Applied Mechanics and Engineering, 389 (2022), p. 114296, https://doi.org/https://doi.org/10.1016/j.cma.2021.114296. 2

[112] D. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Transactions on Automatic Control, 13 (1968), pp. 114–115, https://doi.org/10.1109/TAC.1968.1098829. 131

[113] B. Kramer and A. A. Gorodetsky, *System identification via CUR-factored Hankel approximation*, SIAM Journal on Scientific Computing, 40 (2018), pp. A848–A866, https://doi.org/10.1137/17M1137632. 9, 44

[114] B. Kramer and K. E. Willcox, *Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition*, AIAA Journal, 57 (2019), pp. 2297–2307, https://doi.org/10.2514/1.J057791. 24

[115] S. Lefteriu and A. C. Antoulas, *A new approach to modeling multiport systems from frequency-domain data*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29 (2010), pp. 14–27. 29

[116] S. Lefteriu, A. C. Antoulas, and A. C. Ionita, *Parametric model reduction in the Loewner framework**, IFAC Proceedings Volumes, 44 (2011), pp. 12751–12756, https://doi.org/https://doi.org/10.3182/20110828-6-IT-1002.02651. 18th IFAC World Congress. 98

[117] S. Lefteriu, A. C. Ionita, and A. C. Antoulas, *Modeling Systems Based on Noisy Frequency and Time Domain Measurements*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 365–378, https://doi.org/10.1007/978-3-540-93918-4_33. 29

[118] K. K. Lin and F. Lu, *Data-driven model reduction, Wiener projections, and the Koopman-Mori-Zwanzig formalism*, Journal of Computational Physics, 424 (2021), p. 109864, https://doi.org/https://doi.org/10.1016/j.jcp.2020.109864. 2

[119] P. Lopes dos Santos, J. A. Ramos, and J. L. Martins de Carvalho, *Identification of bilinear systems with white noise inputs: An iterative deterministic-stochastic subspace approach*, IEEE Transactions on Control Systems Technology, 17 (2009), pp. 1145–1153, https://doi.org/10.1109/TCST.2008.2002041. xvi, 2, 101, 103, 117, 118, 119

[120] E. Lorenz, *Deterministic nonperiodic flow*, Journal of the Atmospheric Sciences, 20 (1963), pp. 130–141. 135

[121] M. W. Mahoney and P. Drineas, *CUR matrix decompositions for improved data analysis*, Proceedings of the National Academy of Sciences, 106 (2009), pp. 697–702, https://doi.org/10.1073/pnas.0803205106. 9

[122] A. Mayo and A. Antoulas, *A framework for the solution of the generalized realization problem*, Linear Algebra and its Applications, 425 (2007), pp. 634 – 662, https://doi.org/https://doi.org/10.1016/j.laa.2007.03.008. Special Issue in honor of Paul Fuhrmann. 2, 28, 29, 31, 33

[123] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, *The AAA algorithm for rational approximation*, SIAM Journal on Scientific Computing, 40 (2018), pp. A1494–A1522, https://doi.org/10.1137/16M1106122. 2, 28, 29, 57

[124] V. Olshevsky, ed., *Structured matrices in mathematics, computer science, and engineering. I*, vol. 280 of Contemporary Mathematics, American Mathematical Society, Providence, RI, 2001. 9, 44

[125] I. Oseledets and E. Tyrtyshnikov, *TT-cross approximation for multidimensional arrays*, Linear Algebra and its Applications, 432 (2010), pp. 70 – 88, https://doi.org/https://doi.org/10.1016/j.laa.2009.07.024. 44

[126] B. K. Patel, *Application of Loewner framework for data-driven modeling and diagnosis of polymer electrolyte membrane fuel cells*, 2023. 29

[127] B. Peherstorfer, S. Gugercin, and K. Willcox, *Data-driven reduced model construction with time-domain Loewner models*, SIAM Journal on Scientific Computing, 39 (2017), pp. A2152–A2178, https://doi.org/10.1137/16M1094750. 3, 28, 29, 68

[128] B. Peherstorfer and K. Willcox, *Data-driven operator inference for nonintrusive projection-based model reduction*, Computer Methods in Applied Mechanics and Engineering, 306 (2016), pp. 196–215, https://doi.org/10.1016/j.cma.2016.03.025. 2, 146

[129] M. Petkovska, D. Nikolić, and A. Seidel-Morgenstern, *Nonlinear frequency response method for evaluating forced periodic operations of chemical reactors*, Israel Journal of Chemistry, 58 (2018), pp. 663–681, https://doi.org/10.1002/ijch.201700132. 18

[130] J. Phillips, *Projection-based approaches for model reduction of weakly nonlinear, time-varying systems*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 22 (2003), pp. 171–187, https://doi.org/10.1109/TCAD.2002.806605. 69

[131] I. Pontes Duff, P. Goyal, and P. Benner, *Data-Driven Identification of Rayleigh-Damped Second-Order Systems*, Springer International Publishing, Cham, 2022, pp. 255–272, https://doi.org/https://doi.org/10.1007/978-3-030-95157-3_14. 29

[132] C. Poussot-Vassal, P. Vuillemin, O. Cantinaud, and F. Sève, *Interpolatory methods for generic bizjet gust load alleviation function*, SIAM Journal on Applied Dynamical Systems, 20 (2021), pp. 2391–2411, https://doi.org/10.1137/20M1384014, https://doi.org/10.1137/20M1384014. 29

[133] J. L. Proctor, S. L. Brunton, and J. N. Kutz, *Dynamic mode decomposition with control*, SIAM Journal on Applied Dynamical Systems, 15 (2016), pp. 142–161, https://doi.org/10.1137/15M1013857. 2

[134] D. Quero, P. Vuillemin, and C. Poussot-Vassal, *A generalized eigenvalue solution to the flutter stability problem with true damping: The p-l method*, Journal of Fluids and Structures, 103 (2021), p. 103266, https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2021.103266. 29

[135] P. Rapisarda and A. Antoulas, *A duality perspective on Loewner rational interpolation and state-space modelling of vector-exponential trajectories*, in 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 2096–2100, https://doi.org/10.1109/CDC.2015.7402516. 29

[136] P. Rapisarda and A. C. Antoulas, *Bilinear differential forms and the Loewner framework for rational interpolation*, in Mathematical Control Theory II, M. N. Belur, M. K. Camlibel, P. Rapisarda, and J. M. Scherpen, eds., Cham, 2015, Springer International Publishing, pp. 23–43, https://doi.org/https://doi.org/10.1007/978-3-319-21003-2_2. 29

[137] W. J. Rugh, *Nonlinear system theory - The Volterra/Wiener approach*, University Press, 1981. 17, 83, 85, 87, 89, 123, 175

[138] T. Rüther, I. V. Gosea, L. Jahn, A. C. Antoulas, and M. A. Danzer, *Introducing the Loewner method as a data-driven and regularization-free approach for the distribution of relaxation times analysis of lithium-ion batteries*, Batteries, 9 (2023), https://doi.org/10.3390/batteries9020132, https://www.mdpi.com/2313-0105/9/2/132. 29

[139] S. Sastry, *Nonlinear Systems: Analysis, Stability and Control*, Springer, New York, 1999. 17

[140] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, Journal of Fluid Mechanics, 656 (2010), p. 5–28, https://doi.org/10.1017/S0022112010001217. 2

[141] P. Schulze and B. Unger, *Data-driven interpolation of dynamical systems with delay*, Syst. Control. Lett., 97 (2016), pp. 125–131. 29

[142] J. D. Simard and A. Astolfi, *An interconnection-based interpretation of the Loewner matrices*, in 2019 IEEE 58th Conference on Decision and Control (CDC), 2019, pp. 7788–7793, https://doi.org/10.1109/CDC40024.2019.9030223. 29

[143] J. D. Simard and A. Astolfi, *Loewner functions for linear time-varying systems with applications to model reduction*, IFAC-PapersOnLine, 53 (2020), pp. 5623–5628, https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.1578. 21st IFAC World Congress. 29

[144] T. SIU and M. SCHETZEN, *Convergence of Volterra series representation and bibo stability of bilinear systems*, International Journal of Systems Science, 22 (1991), pp. 2679–2684, https://doi.org/10.1080/00207729108910824. 77

[145] D. Sorensen and A. Antoulas, *The Sylvester equation and approximate balanced reduction*, Linear Algebra and its Applications, 351-352 (2002), pp. 671 – 700, https://doi.org/https://doi.org/10.1016/S0024-3795(02)00283-5. Fourth Special Issue on Linear Systems and Control. 15

[146] D. C. Sorensen and M. Embree, *A deim induced CUR factorization*, SIAM Journal on Scientific Computing, 38 (2016), pp. A1454–A1482, https://doi.org/10.1137/140978430. 9, 44

[147] A. Sorrentino, B. Patel, I. V. Gosea, A. C. Antoulas, and T. Vidaković-Koch, *Determination of the distribution of relaxation times through Loewner framework: A direct and versatile approach*, Journal of Power Sources, 585 (2023), p. 233575, https://doi.org/https://doi.org/10.1016/j.jpowsour.2023.233575. 29

[148] R. Toth, H. S. Abbas, and H. Werner, *On the state-space realization of lpv input-output models: Practical approaches*, IEEE Transactions on Control Systems Technology, 20 (2012), pp. 139–153, https://doi.org/10.1109/TCST.2011.2109040. 2

[149] L. N. Trefethen, *Approximation Theory and Approximation Practice (Other Titles in Applied Mathematics)*, Society for Industrial and Applied Mathematics, USA, 2012. 29, 53

[150] Verspecht and D. Root, *Polyharmonic distortion modeling*, IEEE Microwave Magazine, 7 (2006), pp. 44–57, https://doi.org/10.1109/MMW.2006.1638289. 123, 129, 139, 166

[151] C. D. Villemagne and R. E. Skelton, *Model reductions using a projection formulation*, in 26th IEEE Conference on Decision and Control, vol. 26, 1987, pp. 461–466, https://doi.org/10.1109/CDC.1987.272862. 13

[152] J. Weiss, *A Tutorial on the Proper Orthogonal Decomposition*, American Institute of Aeronautics and Astronautics, 2019, ch. chapter, p. pp, https://doi.org/10.2514/6.2019-3333. 2

[153] X. Y. Z. Xiong, L. J. Jiang, J. E. Schutt-Ainé, and W. C. Chew, *Volterra series-based time-domain macromodeling of nonlinear circuits*, IEEE Transactions on Components, Packaging and Manufacturing Technology, 7 (2017), pp. 39–49, https://doi.org/10.1109/TCPMT.2016.2627601. xvii, 166, 167, 168

[154] X. Y. Z. Xiong, L. J. Jiang, J. E. Schutt-Ainé, and W. C. Chew, *Volterra series-based time-domain macromodeling of nonlinear circuits*, IEEE Transactions on Components, Packaging and Manufacturing Technology, 7 (2017), pp. 39–49, https://doi.org/10.1109/TCPMT.2016.2627601. 129, 175

[155] A. Yousuff and R. Skelton, *Covariance equivalent realizations with application to model reduction of large-scale systems*, in Decentralized/Distributed Control and Dynamic Systems, Part 1 of 3, C. LEONDES, ed., vol. 22 of Control and Dynamic Systems, Academic Press, 1985, pp. 273 – 348, https://doi.org/https://doi.org/10.1016/B978-0-12-012722-1.50014-2. 13

[156] A. Yousuff, D. Wagie, and R. Skelton, *Linear system approximation via covariance equivalent realizations*, Journal of Mathematical Analysis and Applications, 106 (1985), pp. 91 – 115, https://doi.org/https://doi.org/10.1016/0022-247X(85)90133-7. 13

[157] B. Zhu, Q. Zhang, Y. Pan, E. M. Mace, B. York, C. A. Antoulas, C. C. Dacso, and B. W. O'Malley, *A cell-autonomous mammalian 12 hr clock coordinates metabolic and stress rhythms*, Cell Metabolism, 6 (2017), pp. 1305–1319, https://doi.org/doi:10.1016/j.cmet.2017.05.004. 29

[158] A. Zuyev and I. V. Gosea, *Approximating a flexible beam model in the Loewner framework*, in 2023 European Control Conference (ECC), 2023, pp. 1–7, https://doi.org/10.23919/ECC57647.2023.10178203. 55

[159] Úředníček, Zdenêk, *Nonlinear systems - describing functions analysis and using*, MATEC Web Conf., 210 (2018), p. 02021, https://doi.org/10.1051/matecconf/201821002021. 155

THESES

1. The Loewner framework (LF) is a powerful data-driven tool that realizes linear time-invariant systems from time or frequency domain measurements capable of serving the identification and reduction goals.

2. The LF method constructs the system's invariant transfer function directly from input-output (time or frequency) measurements with the flexibility of allocating these measurements appropriately for achieving different goals. For instance, when the left and right measurement sets are almost the same, the LF provides rational approximants that satisfy approximately the first optimality conditions (Hermitian interpolation) and asserts a connection with the optimal intrusive method known as iterative rational Krylov algorithm (IRKA).

3. We review the LF as a rational interpolation method for approximating non-rational transfer functions derived from PDEs or challenging approximation theory benchmarks.

4. The LF, in connection with the Volterra series, has been extended to several nonlinear control systems classes such as bilinear, quadratic, quadratic-bilinear, and polynomial with the formulation of the regular Volterra kernels that can be derived with the variational approach. These approaches' limitations start when the model to be discovered or reduced remains unknown, and only simulations can be performed for data acquisition. These LF variants construct reduced models only with direct numerical simulation from a prior accessible high-fidelity model.

5. The Volterra kernels that can be measured from time-domain data acquisition processes are symmetric and can be derived with the growing exponential approach known as the probing method. Thus, the novelty of this thesis was to develop methods that use the Loewner framework with the symmetric Volterra kernels that can be measured from a physical measurement setup. The Loewner framework faced a fundamental problem in directly generalizing within the structure of the symmetric kernels.

6. In the linear case, a multi-harmonic excitation in the time domain maps a bijective frequency map through the Fourier spectrum. Thus, harmonic indexing in the linear case is straightforward, and frequency domain measurements can be derived directly. Consequently, the LF method uses these measurements to realize the dynamics with a linear time-invariant state-space model.

7. In the nonlinear case, exciting the system for data acquisition with multi-tone input produces a complex Fourier spectrum with commensurate frequencies that is a challenging task for harmonic indexing and kernel separation. In this thesis, we provided ways for kernel separation under multi-harmonic excitation. Moreover, the theoretical establishment of the X-parameters machinery that can provide measurements in a real application environment under polyharmonic distortion in connection with nonlinear systems and the Volterra series seems a promising future research direction that has not yet matured.

8. Assumptions on the model structure (e.g., bilinear, quadratic, etc.) can be held from the specific engineering application. For instance, a physical choice of modeling with bilinear systems can be derived from the first principles when inputs multiply the states (inflow-outflow dynamics). When the engineering application concerns flow problems such as Navier-Stokes and Burgers's from first principles, quadratic models must be introduced to cover the dynamics. Cubic systems must be considered for oscillatory examples with nonlinear stiffness or damping (Van der Pol or Duffing oscillator, etc.).

9. The main novelty in this thesis was to construct the nonlinear models $\boldsymbol{\Sigma}$ listed below by having access to input-$u(t)$ and output-$y(t)$ time-domain data.

$$u(t) \rightarrow \boxed{\boldsymbol{\Sigma}: \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathcal{F}(\mathbf{x}(t)) + \mathcal{G}(\mathbf{x}(t))u(t) + \mathbf{B}u(t), \\ y(t) = \mathbf{C}\mathbf{x}(t),\ \mathbf{x}_0 = \mathbf{0},\ t \geq 0. \end{cases}} \rightarrow y(t)$$

   - Bilinear: $\mathcal{F}(x(t)) = \mathbf{0},\ \mathcal{G}(\mathbf{x}(t)) = \mathbf{N}\mathbf{x}(t)$.
   - Quadratic: $\mathcal{F}(x(t)) = \mathbf{Q}(\mathbf{x}(t) \otimes \mathbf{x}(t)),\ \mathcal{G}(\mathbf{x}(t)) = \mathbf{0}$.
   - Quadratic-bilinear: $\mathcal{F}(x(t)) = \mathbf{Q}(\mathbf{x}(t) \otimes \mathbf{x}(t)),\ \mathcal{G}(\mathbf{x}(t)) = \mathbf{N}\mathbf{x}(t)$.

   As detailed in the thesis, when the above structures were derived from the first principles, the methods served the reduction and identification goals well.

10. In this thesis, we also studied nonlinear approximations either with Carleman's bilinearization scheme or with nonlinear embeddings, such as the quadratization lifting procedure. Although the results gave fair approximations, the methods developed in this thesis used mainly the minimality of the linear operator $\mathbf{A}$ from the general structure above. Consequently, when Carleman's bilinearization or lifting strategies were introduced, the dimension of the states was increased, and adaptation of the lifted degree was left as an open problem along with the minimality of nonlinear systems.

11. Finally, machine learning techniques in combination with data engineering methods such as those presented in this thesis can be advantageous. In particular, neural networks (NNs), due to their power to learn input-output maps through the universal approximation theorem, can replace the actual plant with a black box model for generating more data efficiently and within the interpolation regime that NNs

remain reliable. Consequently, utilizing the methods in this thesis with the data generated from NNs, interpretable (state-space) models with theoretical properties such as stability can be used in the extrapolation (prediction, forecasting) regime, making the ad hoc engineering processes more reliable.

STATEMENT OF SCIENTIFIC COOPERATIONS

This work is based on articles and reports (published and unpublished) that have been obtained in cooperation with various coauthors. To guarantee a fair assessment of this thesis, this statement clarifies the contributions that each coauthor has made. The following people contributed to the content of this work:

- Dr. Ion Victor Gosea (DRI member) conceptualized and provided technical (co-writing, validating, coding) support to our shared contributions.
- Mr. Leonidas Gkimisis (CSC member) conceptualized the topic of nonlinear dynamics that exhibits bifurcations to our shared contribution.
- Ms. Kirandeep Kour (CSC member) validated using neural networks to our shared contribution.

EHRENERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert oder verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadenersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, 08.03.2023

Dimitrios S. Karachalios

DECLARATION OF HONOR

I hereby declare that I produced this thesis without prohibited assistance and that all sources of information that were used in producing this thesis, including my own publications, have been clearly marked and referenced.

In particular I have not willfully:

- Fabricated data or ignored or removed undesired results.
- Misused statistical methods with the aim of drawing other conclusions than those warranted by the available data.
- Plagiarized data or publications or presented them in a distorted way.

I know that violations of copyright may lead to injunction and damage claims from the author or prosecution by the law enforcement authorities.

This work has not previously been submitted as a doctoral thesis in the same or a similar form in Germany or in any other country. It has not previously been published as a whole.

Magdeburg, 08.03.2023

Dimitrios S. Karachalios