Frequency-Domain Learning of Dynamical Systems From Time-Domain Data

Michael S. Ackermann

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mathematics

Serkan Gugercin, Chair

Mark Embree

Christopher Beattie

Jeff Borggaard

May 6, 2022

Blacksburg, Virginia

Keywords: Reduced Order Modeling, Numerical Analysis, Data Driven Modeling

Frequency-Domain Learning of Dynamical Systems From Time-Domain Data

Michael S. Ackermann

(ABSTRACT)

Dynamical systems are useful tools for modeling many complex physical phenomena. In many situations, we do not have access to the governing equations to create these models. Instead, we have access to data in the form of input-output measurements. Data-driven approaches use these measurements to construct reduced order models (ROMs), a small scale model that well approximates the true system, directly from input/output data. Frequency domain data-driven methods, which require access to values (and in some cases to derivatives) of the transfer function, have been very successful in constructing high-fidelity ROMs from data. However, at times this frequency domain data can be difficult to obtain or one might have only access to time-domain data. Recently, Burohman et al. [2020] introduced a framework to approximate transfer function values using only time-domain data. We first discuss improvements to this method to allow a more efficient and more robust numerical implementation. Then, we develop an algorithm that performs optimal-$\mathcal{H}_2$ approximation using purely time-domain data; thus significantly extending the applicability of $\mathcal{H}_2$-optimal approximation without a need for frequency domain sampling. We also investigate how well other established frequency-based ROM techniques (such as the Loewner Framework, Adaptive Anderson-Antoulas Algorithm, and Vector Fitting) perform on this identified data, and compare them to the optimal-$\mathcal{H}_2$ model.

Frequency-Domain Learning of Dynamical Systems From Time-Domain Data

Michael S. Ackermann

(GENERAL AUDIENCE ABSTRACT)

Dynamical systems are useful tools for modeling many phenomena found in physics, chemistry, biology, and other fields of science. Dynamical systems are used to model phenomena that evolve in time and respond to inputs, such as the location of a pendulum in motion, the state of an electric grid, vibrations in a beam, the cooling of a heated piece of metal, or any spatially discretized partial differential equation (PDE). Every dynamical system consists of a system of ordinary differential equations (ODEs) that provides the input to state relationship, together with a state to output mapping. For every dynamical system there is an associated transfer function in the frequency domain that directly links inputs to outputs, bypassing the state of the system. In many applications (such as stable flight of a supersonic aircraft), this system of ODEs must be simulated many times per second, but the number of ODEs leads to long (several second) computation time. The process of finding a smaller dynamical system (or, equivalently, a transfer function) that well approximates the original dynamical system and is fast to simulate is called *reduced order modeling*, and is the main application of this thesis.

In this thesis, we consider the situation where we do not know the true dynamical system, but instead can only observe the system's response to an input (such as observing the movement of wind chimes in a breeze of known speed). There are many well established ways to compute reduced order models (ROMs) if one has access to *frequency domain* input-output data, but in this work we assume access to only *time domain* input-output data, which is

typically easier to collect but harder to construct ROMs from. This thesis expands upon a method called the *data informativity framework* introduced by Burohman et al. [2020] to infer values and derivatives of the transfer function using time domain input-output data. The first contribution of this thesis is to provide a robust and efficient implementation for the data informativity framework. We then provide an algorithm for constructing a ROM that is optimal in a frequency domain sense from time domain data. Finally, we investigate how other established frequency domain ROM techniques perform on the learned frequency domain data.

Dedication

To all those who strive to inspire through education.

Acknowledgments

I would like to thank my advisor Dr. Gugercin for his endless support of my research and personal development as a mathematician. Dr. Gugercin has always been willing to take time out of his exceptionally busy days for my questions, and has gone far above the normal requirements of an advisor. I also thank Dr. Arnold, Dr. Haskell, Dr. Gugercin, and the many other professors I had in undergraduate math classes whose passion for their courses directly contributed to my switching my undergraduate major to mathematics. My parents, Teresa and Jim, and my siblings, Colleen and Rob, also have my heartfelt thanks for always encouraging me to do my best in everything I do, and providing me with a love of mathematics from an early age. Finally, I thank my friends for their love and support, and for putting up with my insatiable need to teach them cool math facts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces several concepts that are key to this thesis, including linear discrete time dynamical systems, reduced order modeling of such systems, and measures of how well the reduced order models approximate the true systems. Throughout, motivation for the work is provided.

1.1 Linear Discrete Time Dynamical Systems

Many physical systems can be accurately described by dynamical systems, which usually arise through discretization of PDEs. Dynamical system models of physical systems allow us to analyze behavior of the system and model their responses to various inputs. In this section, we present linear time invariant discrete time dynamical systems, which are dynamical systems that contain no nonlinear interactions and are also discretized in time.

Linear discrete-time dynamical systems have the form

$$\mathcal{S}: \begin{cases} \mathbf{E}\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{b}u[k] \\ \qquad y[k] = \mathbf{c}^\top \mathbf{x}[k] + \mathbf{d}u[k], \end{cases} \tag{1.1}$$

where $\mathbf{E} \in \mathbb{R}^{n\times n}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{c}^\top \in \mathbb{R}^{1\times n}, \mathbf{d} \in \mathbb{R}$ are the state-space matrices; $n \in \mathbb{N}$ is the system dimension; $\mathbf{x}[k] \in \mathbb{R}^n$ is the state at time k; $u[k] \in \mathbb{R}$ is the input at time k;

and $y[k] \in \mathbb{R}$ is the output at time k.

Under the assumption that the initial condition $\mathbf{x}[0] = \mathbf{0}$, we can take the Z-transform of (1.1) to obtain the associated transfer function

$$H(z) = \mathbf{c}^\top (z\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} + \mathbf{d}, \tag{1.2}$$

which provides a direct input-output mapping that does not rely on a state vector, namely

$$\hat{y}(z) = H(z)\hat{u}(z), \tag{1.3}$$

where $\hat{u}$ and $\hat{y}$ are the Z-transforms of the input and output, respectively. For more information on the Z-transform, see [32]. The transfer function $H(z)$ is a rational function in z, i.e., there exist degree n polynomials $Q(z)$ and $P(z)$ such that

$$H(z) = \frac{Q(z)}{P(z)} = \frac{q_n z^n + q_{n-1}z^{n-1} + \cdots + q_1 z + q_0}{z^n + p_{n-1}z^{n-1} + \cdots + p_1 z + p_0}. \tag{1.4}$$

We say that the linear system (1.1) is *asymptotically stable* if the spectrum of the matrix pencil $\lambda\mathbf{E} - \mathbf{A}$ is contained in the open unit disc. We say (1.1) is *unstable* if there is an eigenvalue of the matrix pencil $\lambda\mathbf{E} - \mathbf{A}$ outside the closed unit disc, or if λ_i is a defective eigenvalue on the unit circle. Note that the spectrum of $\lambda\mathbf{E} - \mathbf{A}$ are the poles of (1.4), so an equivalent definition is that a system is asymptotically stable if and only if all poles of (1.4) are contained in the open unit circle.

The system matrices $\mathbf{E}, \mathbf{A}, \mathbf{b}, \mathbf{c}^\top$ are not unique. For any invertible matrices $\mathbf{T}, \mathbf{S} \in \mathbb{C}^{n\times n}$, the matrices $\mathbf{SET}, \mathbf{SAT}, \mathbf{Sb}, \mathbf{c}^\top\mathbf{T}$ form an equivalent description of the system $\mathcal{S}$, i.e.,

$$\mathbf{c}^\top (z\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} = \mathbf{c}^\top\mathbf{T}(z\mathbf{SET} - \mathbf{SAT})^{-1}\mathbf{Sb}.$$

Some system quantities are invariant to the chosen realization. The Hankel singular values of a system are one such quantity. Hankel singular values are the non-zero singular values of the underlying Hankel operator of $\mathcal{S}$. Each order n system has at most n non-zero Hankel singular values, and their decay rate provides a measure of how many states are required to accurately model the system's behavior [3]. If the Hankel singular values decay quickly, then the system can be approximated well with a system of relatively small order. If the Hankel singular values do not decay quickly, then we may struggle to find good reduced order models. For more information on the Hankel operator and its role in reduced order modeling, see [3]. The concept of a reduced order model is introduced in Section 1.3, while methods for constructing reduced order models are introduced in Chapter 2

1.2 Notation

Throughout this thesis, we use the following notation.

Table 1.1: Notation

$\mathbb{R}^{m\times n}$	matrices of size $m \times n$ with only real entries
$\mathbb{C}^{m\times n}$	matrices of size $m \times n$ with complex entries
σ	any complex number
$\overline{\sigma}$	the complex conjugate of σ
$\mathbf{A}^H$	the Hermitian transpose of $\mathbf{A}$
$\mathbf{A}^\top$	the transpose of $\mathbf{A}$
$\|\mathbf{x}\|$	the 2-norm of $\mathbf{x}$ ($\|\mathbf{x}\|_2$)
$\mathbf{I}$	the Identity matrix of appropriate size
$\mathbf{i}$	the imaginary unit, $\sqrt{-1}$
$H(z)$ or H	a transfer function for a system. This always refers to the *true* transfer function
$H(\sigma)$	the transfer function $H(z)$ evaluated at $z = \sigma$. This always refers to the *true* value of $H(\sigma)$
$H_r(z)$ or H_r	a reduced order transfer function with order r
$\tilde{H}_r(z)$	a reduced order transfer function constructed from frequency data learned from time domain data
$M_0(\sigma)$	an approximation to $H(\sigma)$ from time domain data
$M_1(\sigma)$	an approximation to $H'(\sigma)$ from time domain data
$\kappa_2(\mathbf{A})$	the 2-norm condition number of $\mathbf{A}$
$\mathbf{x}[k]$	the state vector at time k
$u[k]$	the input at time k
$y[k]$	the output at time k

1.3 Reduced Order Modeling

Creating a linear system model of the form (1.1) requires knowledge of equations that describe the system. For many complex systems, such knowledge is not available. Even in the case when a governing PDE is known, discretization of the PDE in space can cause the dimension n of the resulting linear system to be large enough to effect computational performance during

simulation. Therefore, methods to find reduced order models (ROMs) directly from measured system responses to known inputs are valuable tools to model complex physical systems for which the governing equations are unknown or prohibitively complex. For references on where these problems arise and more information on reduced order modeling, see [2, 3, 7, 8, 18].

The reduced order modeling problem is to find $r \ll n$ and $\mathbf{E}_r \in \mathbb{R}^{r\times r}$, $\mathbf{A}_r \in \mathbb{R}^{r\times r}$, $\mathbf{b}_r \in \mathbb{R}^r$, $\mathbf{c}_r^\top \in \mathbb{R}^{1\times r}$ such that (1.1) can be well approximated by

$$\mathcal{S}_r : \begin{cases} \mathbf{E}_r\mathbf{x}_r[k+1] = \mathbf{A}_r\mathbf{x}_r[k] + \mathbf{b}_r u[k] \\ y[k] = \mathbf{c}_r^\top \mathbf{x}_r[k], \end{cases} \tag{1.5}$$

in some sense (see Section 1.4). These methods are called *data driven* when they rely on observations of the system response and assume no knowledge of the true system matrices $\mathbf{E} \in \mathbb{R}^{n\times n}, \mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{b} \in \mathbb{R}^n, \mathbf{c}^\top \in \mathbb{R}^{1\times n}$. There are many well established methods to compute ROMs from frequency domain data $H(\sigma)$ (Chapter 2), but such data can be costly or impossible to obtain.

There are also ROM methods that work with only data from the time domain, which can be easier to obtain. Some of these methods work only with the time domain input-output data $(\mathbb{U}, \mathbb{Y})$ [10, 20, 24, 29, 33, 34, 37], while others seek to find transfer function data $(H(\sigma))$ from time domain data $(\mathbb{U}, \mathbb{Y})$ [11, 31], then use frequency domain ROM techniques (Chapter 2).

In this thesis, we present an algorithm to estimate frequency information $H(\sigma)$ and $H'(\sigma)$ from time domain input-ouptut data $(\mathbb{U}, \mathbb{Y})$. This algorithm requires only one input trajectory and associated output trajectory (although improvements with more information are possible). This data can then be used to construct ROMs using proven frequency domain techniques.

1.4 Error Measures

When designing ROMs, it is important to have a measure of how well a constructed ROM $\mathcal{S}_r$ of the form (1.5) with transfer function $H_r(z)$ approximates the true system $\mathcal{S}$ of the form (1.1) with transfer function $H(z)$. We do this by determining how large the error transfer function

$$H_{err}(z) = H(z) - H_r(z)$$

is. In this thesis, we use two different norms to determine the size of H_{err}. The first is the $\mathcal{H}_2$ norm:

$$\|H\|_{\mathcal{H}_2} = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{\mathbf{i}\omega})|^2 d\omega}. \tag{1.6}$$

The second measure is the $\mathcal{H}_\infty$ norm:

$$\|H\|_{\mathcal{H}_\infty} = \max_{\omega\in[0,2\pi]} |H(e^{\mathbf{i}\omega})|. \tag{1.7}$$

We remark that we are almost always interested in a relative $\mathcal{H}_2$ or $\mathcal{H}_\infty$ error, i.e., we are interested in the quantity

$$\frac{\|H - H_{err}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}} \quad \text{or} \quad \frac{\|H - H_{err}\|_{\mathcal{H}_\infty}}{\|H\|_{\mathcal{H}_\infty}}. \tag{1.8}$$

For more detail, see [3, 44].

Chapter 2

Reduced Order Modeling Methods

This chapter introduces four data driven reduced order modeling algorithms, the Loewner Framework [28], Vector Fitting [19, 38], the Adaptive Anderson-Antoulas Algorithm (AAA) [30], and the Iterative Rational Krylov Algorithm (IRKA) [5, 18]. Specifically, the Realization Independent (TF)-IRKA [5] Algorithm is discussed. Each of these methods uses frequency information and will be revisited later in Chapter 5. Only discussion that is relevant to the rest of this work is provided; for more information see the cited works.

2.1 Loewner Framework

The Loewner Interpolation Framework [28] is a data driven reduced order modeling method that assumes access to values and derivatives of a transfer function $H(z)$. Note that $H(z)$ need not be rational, but the Loewner framework produces a rational function $H_r(z) = \mathbf{c}_r^\top(z\mathbf{E}_r - \mathbf{A}_r)^{-1}\mathbf{b}_r$ that interpolates $H(z)$ at chosen points $\sigma_i \in \mathbb{C}$. The presentation and discussion in this section follows that of [2]. For simplicity, we assume that each σ_i is complex (no σ_i is purely real); see [2] for a full discussion.

Theorem 2.1. *For given distinct points $\{\sigma_i\}_{i=1}^r \in \mathbb{C} \setminus \mathbb{R}$, closed under conjugation, assume*

access to $\{H(\sigma_i)\}_{i=1}^r$ *and* $\{H'(\sigma_i)\}_{i=1}^r$. *Construct the matrix* $\mathbb{L} \in \mathbb{C}^{r\times r}$ *by*

$$\mathbb{L}_{ij} = \begin{cases} -\frac{H(\sigma_i)-H(\sigma_j)}{\sigma_i-\sigma_j}, & \text{if } i \neq j; \\ -H'(\sigma_i), & \text{if } i = j. \end{cases} \tag{2.1}$$

Construct the matrix $\mathbb{M} \in \mathbb{C}^{r\times r}$ *by*

$$\mathbb{M}_{ij} = \begin{cases} -\frac{\sigma_i H(\sigma_i)-\sigma_j H(\sigma_j)}{\sigma_i-\sigma_j}, & \text{if } i \neq j; \\ -(H(\sigma_i)+\sigma_i H'(\sigma_i)), & \text{if } i = j. \end{cases} \tag{2.2}$$

Construct the vectors $\mathbb{Z} \in \mathbb{C}^r$ *and* $\mathbb{Y}^\top \in \mathbb{C}^{1\times r}$ *by*

$$\mathbb{Z}_i = \mathbb{Y}_i^\top = H(\sigma_i). \tag{2.3}$$

If $\mathbb{M}-\sigma_i\mathbb{L}$ *is invertible for each* $i = 1, \ldots, r$, *then the transfer function* $\hat{H}_r(z) = \mathbb{Y}^\top(z\mathbb{L}-\mathbb{M})^{-1}\mathbb{Z}$ *interpolates* $H(z)$ *and* $H'(z)$ *at each* σ_i, *that is,*

$$\hat{H}_r(\sigma_i) = H(\sigma_i), \qquad i = 1, 2, \ldots, r, \text{ and} \tag{2.4}$$

$$\hat{H}_r'(\sigma_i) = H'(\sigma_i), \qquad i = 1, 2, \ldots, r. \tag{2.5}$$

There are two main practical extensions with Loewner interpolants. The first is finding a realization for the reduced transfer function $H_r(z) = \mathbf{c}_r^\top(z\mathbf{E}_r - \mathbf{A}_r)^{-1}\mathbf{b}_r$ that has only real entries in each $\mathbf{E}_r, \mathbf{A}_r, \mathbf{b}_r, \mathbf{c}_r$. For a system with real transfer function $H(z)$, i.e., $H(\sigma) = \overline{H(\overline{\sigma})}$, the resulting reduced order models are real as well. If a transfer function is real, then it has a realization where the entries in each of $\mathbf{E}_r, \mathbf{A}_r, \mathbf{b}_r, \mathbf{c}_r$ are real. It is clear from Theorem 2.1 that the realization for $\hat{H}_r(z)$ produced by Loewner will not necessarily have real entries in all $\mathbf{E}_r, \mathbf{A}_r, \mathbf{b}_r, \mathbf{c}_r$.

To find a real matrix realization, recall from Section 1.1 that for invertible matrices $\mathbf{T}, \mathbf{S} \in \mathbb{C}^{r\times r}$, $(\mathbf{S}\mathbb{L}\mathbf{T}, \mathbf{S}\mathbb{M}\mathbf{T}, \mathbf{S}\mathbb{Z}, \mathbb{Y}\mathbf{T})$ is an equivalent description of the transfer function $\hat{H}_r$.

Let

$$\mathbf{F} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix} \in \mathbb{C}^{2\times 2} \tag{2.6}$$

and let $\mathbf{T} \in \mathbb{C}^{r\times r}$ be the matrix with $\mathbf{F}$ repeated $\frac{r}{2}$ times on the diagonal. Then

$$\begin{aligned} &\mathbf{E}_r = \mathbf{T}^{-1}\mathbb{L}\mathbf{T} \in \mathbb{R}^{r\times r}, \quad \mathbf{A}_r = \mathbf{T}^{-1}\mathbb{M}\mathbf{T} \in \mathbb{R}^{r\times r}, \\ &\mathbf{b}_r = \mathbf{T}^{-1}\mathbb{Z} \in \mathbb{R}^r, \text{ and } \quad \mathbf{c}_r^\top = \mathbb{Y}^\top\mathbf{T} \in \mathbb{R}^{1\times r} \end{aligned} \tag{2.7}$$

have real entries and the transfer function $H_r(\sigma) = \mathbf{c}_r^\top(\sigma\mathbf{E}_r - \mathbf{A}_r)^{-1}\mathbf{b}_r$ has the same value as $\hat{H}_r(\sigma)$ for all $\sigma \in \mathbb{C}$.

The second extension for Loewner interpolants is removing rank deficiency in the matrix pencil $(\lambda\mathbf{E}_r - \mathbf{A}_r)$, which occurs in the case when our data is redundant. To remove rank deficiencies, let

$$\begin{aligned} \begin{bmatrix} \mathbf{E}_r & \mathbf{A}_r \end{bmatrix} &= \mathbf{Y}\Theta_1\mathbf{X}_1 \qquad \mathbf{Y} \in \mathbb{R}^{r\times r}, \Theta_1 \in \mathbb{R}^{r\times 2r}, \mathbf{X}_1 \in \mathbb{R}^{2r\times 2r} \\ \begin{bmatrix} \mathbf{E}_r \\ \mathbf{A}_r \end{bmatrix} &= \mathbf{Y}_1\Theta_2\mathbf{X} \qquad \mathbf{Y}_1 \in \mathbb{R}^{2r\times 2r}, \Theta_2 \in \mathbb{R}^{2r\times r}, \mathbf{X} \in \mathbb{R}^{r\times r} \end{aligned} \tag{2.8}$$

be singular value decompositions. For some tolerance $\epsilon > 0$, set ρ to be the last singular value greater than $\epsilon\sigma_1$ (equivalently, $\frac{\sigma_\rho}{\sigma_1} > \epsilon$). Then take $\mathcal{V}$ to be the first ρ columns of $\mathbf{Y}$ and $\mathcal{W}$ to be the first ρ rows of $\mathbf{X}$ and set

$$\begin{aligned} \mathbf{E}_\rho &= \mathcal{V}^T \mathbf{E}_r \mathcal{W} \in \mathbb{R}^{\rho \times \rho}, \quad \mathbf{A}_\rho = \mathcal{V}^T \mathbf{A}_r \mathcal{W} \in \mathbb{R}^{\rho \times \rho}, \\ \mathbf{b}_\rho &= \mathcal{V}^T \mathbf{b}_r \in \mathbb{R}^\rho, \text{ and } \quad \mathbf{c}_\rho^\top = \mathbf{c}_r^\top \mathcal{W} \in \mathbb{R}^{1 \times \rho}. \end{aligned} \tag{2.9}$$

The transfer function $H_\rho(z) = \mathbf{c}_\rho^\top (z\mathbf{E}_\rho - \mathbf{A}_\rho)^{-1} \mathbf{b}_\rho$ has a nonsingular pencil $(\lambda \mathbf{E}_\rho - \mathbf{A}_\rho)$. If ρ is the true rank of $[\ \mathbf{E}_r\ \mathbf{A}_r]$, then H_ρ interpolates H at $\{\sigma_i\}_{i=1}^r$. If ρ is not the true rank, then H_ρ is an approximate interpolant of H at each σ_i. Again, for the full discussion, see [2].

2.2 Rational Least Squares Fitting

Both Vector Fitting (VF) [19, 38] and the Adaptive Anderson Antoulas Algorithm (AAA) [30] attempt to perform rational least squares fitting by minimizing the objective function

$$\sum_{i=1}^{m} |H(\sigma_i) - H_r(\sigma_i)|^2 \tag{2.10}$$

on provided frequency domain data $H(\sigma_i), i = 1, 2, \ldots, m$. We will briefly describe each method before using them in Chapter 5.

2.2.1 Vector Fitting

Vector Fitting (VF) is an extension of the Sanathanan and Koerner (SK) iteration [36]. The SK iteration seeks to find an order r transfer function

$$H_r(z) = \frac{Q_r(z)}{P_r(z)}, \tag{2.11}$$

where $Q_r(z)$ is a polynomial of degree $r-1$, and $P_r(z)$ is a polynomial of degree r. SK seeks to minimize

$$\sum_{i=1}^{m} |H(\sigma_i) - H_r(\sigma_i)|^2 = \sum_{i=1}^{m} \left| \frac{(Q_r(\sigma_i) - P_r(\sigma_i)H(\sigma_i))}{P_r(\sigma_i)} \right|^2 . \tag{2.12}$$

The optimization function (2.12) is nonlinear, and is replaced with an iteration

$$\sum_{i=1}^{m} \left| \frac{(Q_r^{(k+1)}(\sigma_i) - P_r^{(k+1)}(\sigma_i)H(\sigma_i))}{P_r^{(k)}(\sigma_i)} \right| , \tag{2.13}$$

where $P_r^{(0)} \equiv 1$. The optimization function (2.13) is now linear in its unknowns ($Q_r^{(k+1)}$ and $P_r^{(k+1)}$) for each k.

The contribution of VF is to represent (2.11) in the noninterpolatory barycentric form

$$H_r(z) = \frac{Q_r(z)}{P_r(z)} = \sum_{j=1}^{r} \frac{\mu_j}{z - z_j} \Bigg/ \sum_{j=1}^{r} \frac{\nu_j}{z - z_j} , \tag{2.14}$$

where z_j are the support points. At step k, the coefficients for the $(k+1)^{\text{st}}$ iteration μ_j and ν_j are found by solving (2.13). The support points z_j are updated in a way that enhances stability of the iteration. Note that VF is not interpolating any of the transfer function values $H(\sigma_i)$, but using all $2r$ degrees of freedom to perform a least squares fit to the data. For more information see [13, 14, 19].

There are other rational least squares methods, see e.g. [9, 20, 21]. While convergence of Vector Fitting is not guaranteed [25], in most cases with a good initial guess, Vector Fitting converges quickly.

2.2.2 Adaptive Anderson-Antoulas Algorithm

The Adaptive Anderson-Antoulas (AAA) algorithm [30], like Vector Fitting, also attempts to find a reduced order model by rational least squares fitting. The difference is that, for an order r ROM, AAA also enforces interpolation at r points.

For given frequency information $\mathcal{Z} = \{\sigma_i\}_{i=1}^m$ and $H(\mathcal{Z}) = \{H(\sigma_i)\}_{i=1}^m$ and a given maximum order $\tilde{r}$, AAA seeks an order $r \leq \tilde{r}$ transfer function in interpolatory barycentric form

$$H_r(z) = \frac{Q_r(z)}{P_r(z)} = \sum_{j=1}^{r} \frac{w_j f_j}{z - z_j} \Bigg/ \sum_{j=1}^{r} \frac{w_j}{z - z_j} ,$$

where $w_j \neq 0$. For each $r = 1, 2, \ldots, \tilde{r}$, AAA enforces interpolation at r points $\{\sigma_{i_j}\}_{j=1}^r \subset \mathcal{Z}$ by setting

$$z_j = \sigma_{i_j}, \quad f_j = H(\sigma_{i_j}), \quad j = 1, 2, \ldots, r.$$

This leaves an extra set of parameters, the weights w_j, to perform least squares fitting at points where interpolation is not enforced. The points for interpolation at iteration r are the $r - 1$ points previously interpolated, as well as the point $\sigma_i \in \mathcal{Z}$, where

$$|H_{r-1}(\sigma_i) - H(\sigma_i)|$$

is maximized. Finally, the vector of weights $w = [w_1, w_2 \ldots, w_r]^\top \in \mathbb{R}^r$ are chosen by linearizing

$$\min_{w \in \mathbb{R}^r, \|w\|=1} \sum_{i \neq i_j} \left(H(\sigma_i) - \frac{Q_r(\sigma_i)}{P_r(\sigma_i)} \right)^2 \tag{2.15}$$

as

$$\min_{w \in \mathbb{R}^r, \|w\|=1} \sum_{i \neq i_j} \left(H(\sigma_i) P_r(\sigma_i) - Q_r(\sigma_i) \right)^2 . \tag{2.16}$$

The indices of the interpolation points i_j are left out of the sum (2.16) since we enforce

interpolation at each σ_{i_j}. AAA converges when the nonlinear residual

$$\sum_{i \neq i_j} \left(H(\sigma_i) - \frac{Q_r(\sigma_i)}{P_r(\sigma_i)} \right)^2 \tag{2.17}$$

satisfies a given tolerance, or we reach our maximum allowed approximation order.

AAA has applications outside of reduced order modeling, such as nonlinear eigenvalue problems [26], rational minimax problems [16], and rational interpolation over disconnected domains [30]. AAA has been extended to parametric systems [35] and systems with quadratic output, [17]. A variant of AAA that gives up the barycentric representation in favor of enforcing stability of the ROM is presented in [39].

2.3 Iterative Rational Krylov Algorithm

The Iterative Rational Krylov Algorithm (IRKA) [18] attempts to find a locally $\mathcal{H}_2$ optimal ROM from given starting shifts (or interpolation points). This section provides the criteria for a reduced system to be $\mathcal{H}_2$ optimal, and gives an algorithmic description of the Realization Independent IRKA (TF-IRKA) [5].

2.3.1 $\mathcal{H}_2$ optimality conditions

An asymptotically stable reduced system $\mathcal{S}_r$ as in (1.5) with associated transfer function $H_r(z)$ is said to be an $\mathcal{H}_2$ optimal approximation of a system $\mathcal{S}$ as in (1.1) with associated transfer function $H(z)$ if

$$\|H - H_r\|_{\mathcal{H}_2} = \min_{\tilde{H}_r} \|H - \tilde{H}_r\|_{\mathcal{H}_2}, \tag{2.18}$$

with $\tilde{H}_r$ ranging over all degree r rational functions. While not immediately apparent, if H_r is an optimal order r approximation to H, then H_r interpolates H and H' at r points in $\mathbb{C}$.

Theorem 2.2. *Let H_r be a locally optimal order r approximation to H in the $\mathcal{H}_2$ norm. Let $\{\lambda_i\}_{i=1}^r$ be the poles of H_r. Then H_r is a Hermite interpolant to H at $\frac{1}{\lambda_i}$. That is, for each $i = 1, \ldots, r$,*

$$\begin{aligned} H_r\left(\frac{1}{\lambda_i}\right) &= H\left(\frac{1}{\lambda_i}\right) \text{ and} \\ H_r'\left(\frac{1}{\lambda_i}\right) &= H'\left(\frac{1}{\lambda_i}\right). \end{aligned} \tag{2.19}$$

For a proof of Theorem 2.2 in a more general context, as well as an extended discussion, see [2].

2.3.2 The Realization Independent Iterative Rational Krylov Algorithm

Theorem 2.2 provides a means to check that we have an $\mathcal{H}_2$ optimal ROM, but does not provide a method for finding one. IRKA [2, 5, 18] in its original formulation computes an $\mathcal{H}_2$ optimal ROM for $\mathcal{S}$ via interpolary projections. These interpolary projections require knowledge of the system matrices, which are not always available. The Loewner framework (Section 2.1) provides a method to construct these interpolants directly from frequency data. Replacing the direct interpolary projection in IRKA with an implicit, data-driven interpolary projection driven by the Loewner framework is known as the Realization Independent Iterative Rational Krylov Algorithm (TF-IRKA) [5], and is described in Algorithm 1.

The main cost of Algorithm 1 is to resample H and H' at every step. One contribution of this thesis is to provide a method to quickly sample H and H' when one only has access to time domain information. Attempts at finding faster ways to produce $\mathcal{H}_2$ optimal ROMs

Algorithm 1 TF-IRKA

Require: $\{\sigma_i\}_{i=1}^r$, an initial set of interpolation points closed under conjugation (i.e., if σ is an interpolation point, so is $\overline{\sigma}$).
Sample H and H' at $\{\sigma_i\}_{i=1}^r$
Form $\mathbf{E}_\rho$ and $\mathbf{A}_\rho$ using σ_i, $H(\sigma_i)$, and $H'(\sigma_i)$ from (2.9).
while Not Converged **do**
    Calculate generalized eigenvalues $\{\lambda_i\}_{i=1}^r$ of the pencil $\lambda\mathbf{E}_\rho - \mathbf{A}_\rho$
    $\sigma_i \leftarrow \frac{1}{\lambda_i}$
    Resample H and H' at new $\{\sigma_i\}_{i=1}^r$
    Form new $\mathbf{E}_\rho$ and $\mathbf{A}_\rho$ from (2.9) using σ_i, $H(\sigma_i)$, and $H'(\sigma_i)$.
end while
Form $\mathbf{E}_\rho, \mathbf{A}_\rho, \mathbf{B}_\rho, \mathbf{C}_\rho^\top$ from (2.9)

have already been made, such as in [22].

Chapter 3

Data Informativity Framework for Moment Matching

This chapter summarizes the results from [11], which provides a method to calculate values and derivatives (moments) of transfer functions of discrete time linear dynamical systems from time domain data based on the data informativity framework [41]. The background and main results are given here. Further discussion and improvements leading to a better numerical implementation, which are some of the main contributions of this thesis, are provided in Chapter 4. For proofs and other discussion, see [11].

Data informativity is also used in [43] to identify controllers for continuous time dynamical systems from data, without first identifying the full model.

3.1 Problem Formulation

Let $\mathcal{S}$ be any order n discrete time SISO system with state-space realization

$$\mathcal{S}: \begin{cases} \mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{b}u[k] \\ y[k+1] = \mathbf{c}^\top \mathbf{x}[k] + \mathbf{d}u[k], \end{cases} \tag{3.1}$$

$$\text{with } \mathbf{A} \in \mathbb{R}^{n\times n},\ \mathbf{b} \in \mathbb{R}^n,\ \mathbf{c}^\top \in \mathbb{R}^{1\times n}, \mathbf{d} \in \mathbb{R}$$

and transfer function

$$H(z) = \mathbf{c}^\top (z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + \mathbf{d}. \tag{3.2}$$

Recall that $H(z)$ is a rational function in z, which we can write as the ratio of two polynomials

$$H(z) = \frac{Q(z)}{P(z)} = \frac{q_n z^n + q_{n-1}z^{n-1} + \cdots + q_1 z + q_0}{z^n + p_{n-1}z^{n-1} + \cdots + p_1 z + p_0}. \tag{3.3}$$

Then if we multiply both sides of (3.3) by $P(z)$ and take the inverse Z-transform, we obtain an order n difference equation

$$y_{t+n} + p_{n-1}y_{t+n-1} + \cdots + p_1 y_{t+1} + p_0 y_t = q_n u_{t+n} + q_{n-1}u_{t+n-1} + \cdots + q_1 u_{t+1} + q_0 u_t, \tag{3.4}$$

where $t = 0, 1, \ldots, T$ for some $T > n \in \mathbb{N}$. Denote the coefficients in (3.4) by

$$\mathbf{p} = [p_0, p_1, \ldots p_{n-1}] \in \mathbb{R}^n \quad \text{and} \quad \mathbf{q} = [q_0, q_1, \ldots q_n] \in \mathbb{R}^{n+1}.$$

The remainder of this section will assume that the coefficients $\mathbf{p}$ and $\mathbf{q}$ are unknown, but that the order of the system n is known. The assumption that n is known will be removed in Section 4.1. Assume further that for some $T \geq n$ we have access to input data

$$\mathbb{U} = [u_0, u_1, \ldots u_T] \in \mathbb{R}^{T+1}$$

and output data

$$\mathbb{Y} = [y_0, y_1, \dots y_T] \in \mathbb{R}^{T+1}$$

generated by the true system $\mathcal{S}$. Under these assumptions, the goal of this section is to learn frequency data $H(\sigma)$ for $\sigma \in \mathbb{C}$ from the data in $\mathbb{U}$ and $\mathbb{Y}$.

To begin, note that taking $t = 0$, (3.4) can be rearranged and written as

$$\begin{bmatrix} \mathbf{q} & -\mathbf{p} \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_n \\ y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = y_n. \tag{3.5}$$

Since $\mathbb{U}$ and $\mathbb{Y}$ are generated by $\mathcal{S}$, $\mathbb{U}$ and $\mathbb{Y}$ must satisfy (3.4) for each $t = 0, 1, \dots, T - n$. Then for each t, we can express (3.4) in the form (3.5). So we have $T - n + 1$ inner products of the form (3.5), which can be written as the linear system

$$\begin{bmatrix} \mathbf{q} & -\mathbf{p} \end{bmatrix} \begin{bmatrix} \mathbb{H}_n(\mathbb{U}) \\ \overline{\mathbb{H}}_n(\mathbb{Y}) \end{bmatrix} = \begin{bmatrix} y_n & y_{n-1} & \dots & y_T \end{bmatrix}, \tag{3.6}$$

where

$$\mathbb{H}_l(X) = \begin{bmatrix} x_0 & x_1 & \dots & x_{T-l} \\ x_1 & x_2 & \dots & x_{T-l+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_l & x_{l+1} & \dots & x_T \end{bmatrix} \in \mathbb{R}^{(l+1)\times(T-l+1)} \tag{3.7}$$

is the Hankel matrix of depth l and $\overline{\mathbb{H}}_l(X)$ is $\mathbb{H}_l(X)$ with the last row removed. We can now

define the condition for when a system with coefficient vectors

$$\tilde{\mathbf{p}} = [\tilde{p}_0, \tilde{p}_1, \dots \tilde{p}_{n-1}]^\top \in \mathbb{R}^n \quad \text{and} \quad \tilde{\mathbf{q}} = [\tilde{q}_0, \tilde{q}_1, \dots \tilde{q}_n]^\top \in \mathbb{R}^{n+1} \tag{3.8}$$

can explain the data in $\mathbb{U}, \mathbb{Y}$.

Definition 3.1. A system with coefficient vectors (3.8) can explain the data $\mathbb{U}, \mathbb{Y}$ if

$$\begin{bmatrix} \tilde{\mathbf{q}} & -\tilde{\mathbf{p}} \end{bmatrix} \begin{bmatrix} \mathbb{H}_n(\mathbb{U}) \\ \overline{\mathbb{H}}_n(\mathbb{Y}) \end{bmatrix} = \begin{bmatrix} y_n & y_{n-1} & \dots & y_T \end{bmatrix}. \tag{3.9}$$

Now, we can define the set of all system coefficients that explain the data.

Definition 3.2. The set of all coefficient vectors that can explain the data in $\mathbb{U}, \mathbb{Y}$ is

$$\Sigma_{\mathbb{U},\mathbb{Y}} = \left\{ \begin{bmatrix} \tilde{\mathbf{q}} & -\tilde{\mathbf{p}} \end{bmatrix} \in \mathbb{R}^{1\times(2n+1)} \mid \text{(3.9) holds} \right\}. \tag{3.10}$$

Recall from (3.3) that the transfer function $H(z)$ is a rational function, i.e., for any $\sigma \in \mathbb{C}$,

$$H(\sigma) = \frac{q_n\sigma^n + q_{n-1}\sigma^{n-1} + \ldots + q_1\sigma + q_0}{\sigma^n + p_{n-1}\sigma^{n-1} + \ldots + p_1\sigma + p_0}, \tag{3.11}$$

which can be rearranged as

$$\begin{aligned} H(\sigma)\cdot(\sigma^n) =& (q_n\sigma^n + q_{n-1}\sigma^{n-1} + \ldots + q_1\sigma + q_0) \\ & - H(\sigma)(p_{n-1}\sigma^{n-1} + \ldots + p_1\sigma + p_0). \end{aligned} \tag{3.12}$$

We call $H(\sigma)$ the *0-th moment* of the system $\mathcal{S}$ at σ. We will use the notation $H(\sigma) = M_0$.

Similar to (3.6), we can rewrite (3.12) as a linear system

$$\begin{bmatrix} \mathbf{q} & -\mathbf{p} \end{bmatrix} \begin{bmatrix} \gamma_n(\sigma) \\ M_0\gamma_{n-1}(\sigma) \end{bmatrix} = M_0\sigma^n \tag{3.13}$$

where

$$\gamma_k(\sigma) = \begin{bmatrix} 1 \\ \sigma \\ \sigma^2 \\ \vdots \\ \sigma^k \end{bmatrix} \in \mathbb{C}^{k+1}. \tag{3.14}$$

We can now define the condition under which a system with coefficient vector $[\tilde{\mathbf{q}} \ -\tilde{\mathbf{p}}]$ has 0-th moment M_0 at σ.

Definition 3.3. A system with coefficient vector $[\tilde{\mathbf{q}} \ -\tilde{\mathbf{p}}]$ has 0-th moment M_0 at σ if

$$\begin{bmatrix} \tilde{\mathbf{q}} & -\tilde{\mathbf{p}} \end{bmatrix} \begin{bmatrix} \gamma_n(\sigma) \\ M_0\gamma_{n-1}(\sigma) \end{bmatrix} = M_0\sigma^n. \tag{3.15}$$

Now, we can define the set of all systems with 0-th moment M_0 at σ.

Definition 3.4. The set of all systems with 0-th moment M_0 at σ is

$$\Sigma^0_{\sigma,M_0} = \left\{ \begin{bmatrix} \tilde{\mathbf{q}} & -\tilde{\mathbf{p}} \end{bmatrix} \in \mathbb{R}^{1\times(2n+1)} \mid (3.15) \text{ holds} \right\}. \tag{3.16}$$

We can also define $M_1 = H'(\sigma)$, the derivative of H at σ. We also use the helpful relation

$$\begin{aligned} Q'(\sigma) &= P(\sigma)H'(\sigma) + H(\sigma)P'(\sigma) \\ &= P(\sigma)M_1 + M_0P'(\sigma). \end{aligned} \tag{3.17}$$

(3.17) can be derived directly from the quotient rule for differentiation. Expanding (3.17) leads to

$$\begin{aligned} nq_n\sigma^{n-1} + (n-1)q_{n-1}\sigma^{n-2} + \ldots + 2q_2\sigma + q_1 & \\ =&(\sigma^n + p_{n-1}\sigma^{n-1} + \ldots + p_1\sigma + p_0)M_1 \\ &+ M_0(n\sigma^{n-1} + (n-1)p_{n-1}\sigma^{n-2} + \ldots + 2p_2\sigma + p_1). \end{aligned} \tag{3.18}$$

Then, isolating the terms of (3.18) without a p_i or q_i coefficient and grouping terms we obtain

$$\begin{aligned} M_1\sigma^n + nM_0\sigma^{n-1} = & \\ q_n n\sigma^{n-1} + q_{n-1}(n-1)\sigma^{n-2} + \ldots + q_2 2\sigma + q_1 & \\ -p_{n-1}(M_0(n-1)\sigma^{n-2} + M_1\sigma^{n-1}) - \ldots - p_1(M_0 + M_1\sigma) - p_0M_1. & \end{aligned} \tag{3.19}$$

Then in (3.13), (3.17) can be written as a linear system

$$\begin{bmatrix} \mathbf{q} & -\mathbf{p} \end{bmatrix} \begin{bmatrix} \gamma_n^{(1)}(\sigma) \\ M_0\gamma_{n-1}^{(1)}(\sigma) + M_1\gamma_{n-1}(\sigma) \end{bmatrix} = \begin{bmatrix} nM_0\sigma^{n-1} + M_1\sigma^n \end{bmatrix}, \tag{3.20}$$

where

$$\gamma_k^{(1)}(\sigma) = \begin{bmatrix} 0 \\ 1 \\ 2\sigma \\ \vdots \\ k\sigma^{k-1} \end{bmatrix} \in \mathbb{C}^{k+1}. \tag{3.21}$$

As in (3.15), (3.20) will hold for any system with coefficients $\tilde{\mathbf{q}} \in \mathbb{R}^n$ and $\tilde{\mathbf{p}} \in \mathbb{R}^{n+1}$ with first moment M_1 at σ. Finally, we define the set of all system coefficients with first moment M_1 at σ by

$$\Sigma^1_{\sigma,M_1} = \left\{ \begin{bmatrix} \tilde{\mathbf{q}} & -\tilde{\mathbf{p}} \end{bmatrix} \in \mathbb{R}^{1\times(2n+1)} \mid \text{(3.20) holds} \right\}. \tag{3.22}$$

It is also possible to calculate M_k, the k-th moment of the system at σ for any $k = 1, 2, \ldots$. For the formula and derivation, see [11].

3.2 Calculation of M_0

In this section, we will define theoretical conditions under which we can calculate a unique transfer function value $H(\sigma) = M_0$ from input-output data $(\mathbb{U}, \mathbb{Y})$, using the ideas developed in Section 3.1. Since we must calculate M_0, but only have access to $(\mathbb{U}, \mathbb{Y})$, we require that all systems that explain the data $(\mathbb{U}, \mathbb{Y})$ must also have transfer function value M_0 at σ.

Definition 3.5. The data $(\mathbb{U}, \mathbb{Y})$ are informative for interpolation at σ if there exists a unique $M_0 \in \mathbb{C}$ such that

$$\Sigma_{\mathbb{U},\mathbb{Y}} \subset \Sigma^0_{\sigma,M_0}. \tag{3.23}$$

Leveraging (3.15) and (3.9) and applying Definition 3.5, we are presented with a method for calculating M_0

Lemma 3.6 (Burohman et al. [11])**.** *Assume access to input data $\mathbb{U}$ and output data $\mathbb{Y}$. Let n be the dimension of the system, let $\sigma \in \mathbb{C}$, and let $H(\sigma) = M_0$ be the value of the true transfer function at σ. Then $\Sigma_{\mathbb{U},\mathbb{Y}} \subset \Sigma^0_{\sigma,M_0}$ if and only if there exists $\xi \in \mathbb{C}^{T-n+1}$ such that*

$$\begin{bmatrix} \mathbb{H}_n(\mathbb{U}) & 0 \\ \mathbb{H}_n(\mathbb{Y}) & -\gamma_n(\sigma) \end{bmatrix} \begin{bmatrix} \xi \\ M_0 \end{bmatrix} = \begin{bmatrix} \gamma_n(\sigma) \\ 0 \end{bmatrix}. \tag{3.24}$$

Analyzing Lemma 3.6, we find conditions for determining the existence and uniqueness of M_0 from $(\mathbb{U}, \mathbb{Y})$.

Theorem 3.7 (Burohman et al. [11])**.** *The data $(\mathbb{U},\mathbb{Y})$ are informative for interpolation at σ if and only if*

$$\operatorname{rank} \begin{bmatrix} \mathbb{H}_n(\mathbb{U}) & 0 & \gamma_n(\sigma) \\ \mathbb{H}_n(\mathbb{Y}) & \gamma_n(\sigma) & 0 \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \mathbb{H}_n(\mathbb{U}) & 0 \\ \mathbb{H}_n(\mathbb{Y}) & \gamma_n(\sigma) \end{bmatrix} \tag{3.25}$$

and

$$\operatorname{rank}\begin{bmatrix}\mathbb{H}_n(\mathbb{U}) & 0\\ \mathbb{H}_n(\mathbb{Y}) & \gamma_n(\sigma)\end{bmatrix} = \operatorname{rank}\begin{bmatrix}\mathbb{H}_n(\mathbb{U})\\ \mathbb{H}_n(\mathbb{Y})\end{bmatrix} + 1. \tag{3.26}$$

The existence of M_0 is given by (3.25), which guarantees that the right hand side of (3.24) is in the range of the matrix on the left hand side. The uniqueness of M_0 is given by (3.26) by guaranteeing that the last column of the matrix in (3.24) is not linearly dependent on the others. This condition indicates that the matrix

$$\begin{bmatrix}\mathbb{H}_n(\mathbb{U})\\ \mathbb{H}_n(\mathbb{Y})\end{bmatrix}$$

cannot have full row rank. The implications of this requirement are explored in Section 4.2.

3.3 Calculation of M_1

In this section, we will define theoretical conditions under which we can calculate a unique transfer function derivative $H'(\sigma) = M_1$ from input-output data $(\mathbb{U}, \mathbb{Y})$, using the ideas developed in Section 3.1. Since we must calculate M_1, but only have access to $(\mathbb{U}, \mathbb{Y})$, we require that all systems that explain the data $(\mathbb{U}, \mathbb{Y})$ must also have transfer function derivative M_1 at σ.

Definition 3.8. The data $(\mathbb{U}, \mathbb{Y})$ are informative for finding M_1 at σ if

1. The data $(\mathbb{U}, \mathbb{Y})$ are informative for interpolation at σ

2. There exists a unique $M_1 \in \mathbb{C}$ such that

$$\Sigma_{\mathbb{U},\mathbb{Y}} \subset \Sigma^1_{\sigma,M_1}$$

Leveraging (3.20) and (3.9) and applying Definition 3.8, we are presented with a method for calculating M_1

Lemma 3.9 (Burohman et al. [11])**.** *Assume access to input data $\mathbb{U}$, output data $\mathbb{Y}$ and the moment M_0 at $\sigma \in \mathbb{C}$. Let n be the dimension of the system, and let $H'(\sigma) = M_1$ be the value of the first derivative of the true transfer function at σ. Then $\Sigma_{\mathbb{U},\mathbb{Y}} \subset \Sigma^1_{\sigma,M_1}$ if and only if there exists $\xi \in \mathbb{C}^{T-n+1}$ such that*

$$\begin{bmatrix} \mathbb{H}_n(\mathbb{U}) & 0 \\ \mathbb{H}_n(\mathbb{Y}) & -\gamma_n(\sigma) \end{bmatrix} \begin{bmatrix} \xi \\ M_1 \end{bmatrix} = \begin{bmatrix} \gamma_n^{(1)}(\sigma) \\ M_0\gamma_n^{(1)}(\sigma) \end{bmatrix}. \tag{3.27}$$

Analyzing Lemma 3.9, we find a condition for determining the existence of M_1 from $(\mathbb{U}, \mathbb{Y})$.

Theorem 3.10 (Burohman et al. [11])**.** *Assume the data $(\mathbb{U}, \mathbb{Y})$ are informative for interpolation at $\sigma \in \mathbb{C}$, with M_0 being the corresponding moment. Then the data $(\mathbb{U}, \mathbb{Y})$ are informative for finding the first moment of $\mathcal{S}$ at σ if and only if*

$$\operatorname{rank} \begin{bmatrix} \mathbb{H}_n(\mathbb{U}) & 0 & \gamma_n^{(1)}(\sigma) \\ \mathbb{H}_n(\mathbb{Y}) & \gamma_n(\sigma) & M_0\gamma_n^{(1)}(\sigma) \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \mathbb{H}_n(\mathbb{U}) & 0 \\ \mathbb{H}_n(\mathbb{Y}) & \gamma_n(\sigma) \end{bmatrix}. \tag{3.28}$$

Note that Theorem 3.10 only supplies an existence condition. This is because by assuming $(\mathbb{U}, \mathbb{Y})$ is informative for interpolation at σ, uniqueness is already guaranteed by Theorem 3.7.

The results presented in this section give us a theoretical means to learn frequency information $H(\sigma)$ given time domain input data $\mathbb{U}$ and output data $\mathbb{Y}$, under the additional

assumption that the system order n is known. Chapter 4 introduces updates to these results that allow for efficient and robust numerical implementation, as well as removes the assumption that n is known.

Chapter 4

Improvements and Numerical Implementation

The results in Chapter 3 from [11] provide theoretical means to calculate the value and first derivative of a transfer function from time domain data. However, from a perspective of numerical implementation, several issues arise, such as the assumed knowledge of the system dimension, ill-conditioned linear systems, and inaccurate calculation of numerical rank. In this chapter, we provide solutions to these issues, which lead to an algorithm that assumes no knowledge about the underlying system.

All results in this chapter are presented for calculation of M_0, but also extend to the calculation of M_1.

4.1 Estimating the system order n

Recall that the theory developed in Section 3.1 had the underlying assumption that we knew the system dimension, n. This assumption causes the data informativity framework to be no longer truly data driven. Numerical experiments show that knowledge of the true system dimension n is not needed, but for some $\hat{n} \leq n$, we can achieve nearly the same accuracy in learned frequency information M_0 as if we knew n.

To show this, for an example system $\mathcal{S}$ as in (1.1) with transfer function $H(z)$ with order $n = 100$, we simulate $\mathcal{S}$ for T time steps to obtain $\mathbb{U} \in \mathbb{R}^{T+1}$ and $\mathbb{Y} \in \mathbb{R}^{T+1}$. Then, using only $\mathbb{U}$ and $\mathbb{Y}$ for each

$$\hat{n} = 2, 4, \ldots 100,$$

we estimate $M_0^{\hat{n}}(\sigma_i) \approx H(\sigma_i)$ at each

$$\sigma_i = e^{\mathbf{i}\omega_i} \in \mathbb{C}, \quad i = 1, 2, \ldots, 100,$$

where $\{\omega_i\}_{i=1}^{100}$ are logarithmically spaced in $[10^{-5}, \pi)$, and $M_0^{\hat{n}}(\sigma_i)$ is the learned value of $H(\sigma_i)$ using (3.24) with $\hat{n}$ in place of n. We also calculate the true transfer function values $\{H(\sigma_i)\}_{i=1}^{100}$. Then the relative error in the calculation of $M_0^{\hat{n}}(\sigma_i)$ is

$$\epsilon_i = \frac{|H(\sigma_i) - M_0^{\hat{n}}(\sigma_i)|}{|H(\sigma_i)|}. \tag{4.1}$$

We recorded

$$\max \epsilon = \max_i \epsilon_i, \quad \text{and} \quad \text{mean } \epsilon = \frac{1}{100} \sum_{i=1}^{100} \epsilon_i.$$

Figure 4.1 shows the results of this experiment for two randomly generated order $n = 100$ stable systems, which we will refer to as $\mathcal{S}_1$ and $\mathcal{S}_2$. The system $\mathcal{S}_1$ has random poles which are complex conjugate pairs, while the system $\mathcal{S}_2$ has random poles which all lie on the real axis. The Hankel singular values of $\mathcal{S}_1$ decay gradually (Figure 4.1c), while the Hankel singular values of $\mathcal{S}_2$ decay more quickly (Figure 4.1d).

We see that while the decay rate in the maximum and mean errors are not equal (Figures 4.1b and 4.1a), neither system requires $\hat{n} = n$ to obtain low relative errors in learned frequency information. Thus, we have shown that we do not need to know n, but rather some $\hat{n} \leq n$ for which the relative error in learned frequency data $M_0^{\hat{n}}$ is small.

An approach to finding a suitable $\hat{n}$ is taken from MIMO Output-Error State Space (MOESEP) model identification [23, 42], presented in Theorem 4.1. While this method is meant for system identification, one step of the method reveals the dimension of the system.

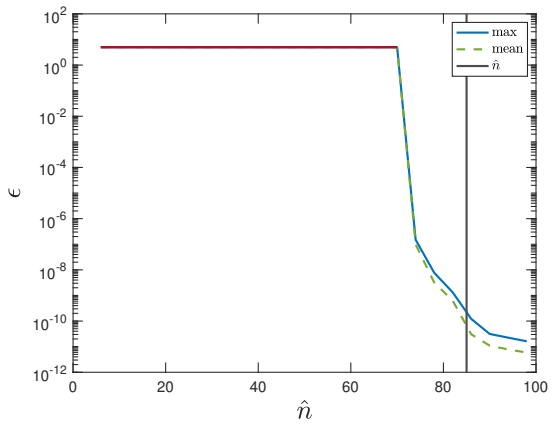
Theorem 4.1. *Given* $\mathbb{U} \in \mathbb{R}^{T+1}$ *and* $\mathbb{Y} \in \mathbb{R}^{T+1}$*, form the Hankel matrices* $\mathbb{H}_k(\mathbb{U})$ *and* $\mathbb{H}_k(\mathbb{Y}) \in \mathbb{R}^{M \times L}$ *as in* (3.7)*, choosing* k *such that* $M, L > n$ *and* $L \geq 2M$*. Assume the underlying system is reachable and that* $\operatorname{rank} \mathbb{H}_k(\mathbb{U}) > n$*. Then let*

$$\begin{bmatrix} \mathbb{H}_k(\mathbb{U})^\top & \mathbb{H}_k(\mathbb{Y})^\top \end{bmatrix} = \mathbf{QR} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & \mathbf{R}_{22} \end{bmatrix} \tag{4.2}$$

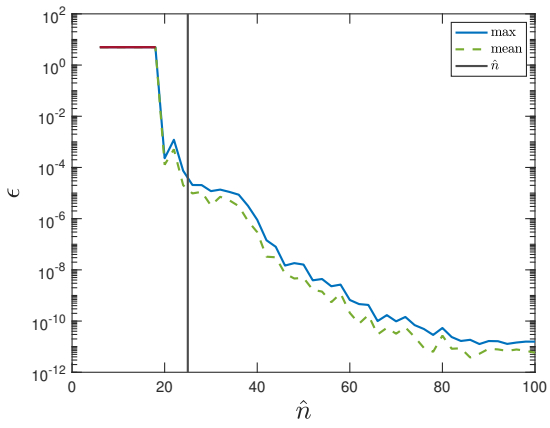
be the short QR decomposition where $\mathbf{R}_{22} \in \mathbb{R}^{M \times M}$*. Then* $\operatorname{rank} \mathbf{R}_{22} = n$*.*

For the proof, see [42]. Note that Theorem 4.1 assumes $M, L > n$. In practice, we collect enough data to make $\mathbb{H}_k(\mathbb{U})$ and $\mathbb{H}_k(\mathbb{Y})$ large enough that we can assume this condition is satisfied. Theorem 4.1 also assumes that $\operatorname{rank} \mathbb{H}_k(\mathbb{U}) > n$. In practice, we assume this condition is met, proceed according to the theorem, then check that the calculated n is indeed greater than $\operatorname{rank} \mathbb{H}_k(\mathbb{U})$.

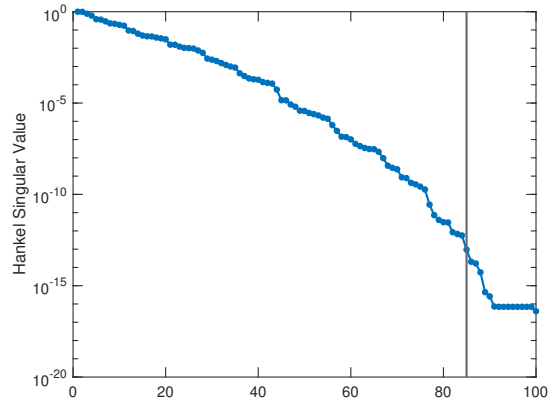
The process given in Theorem 4.1 generally returns some $\hat{n} < n$. The calculated $\hat{n}$ seems to be correlated with the number of (numerically) nonzero Hankel singular values of the true system, and almost always is large enough to provide reasonable accuracy in calculations of frequency information. An example of a system ($\mathcal{S}_1$) where a satisfactory $\hat{n}$ is produced is given in Figure 4.1a, and an example of a system ($\mathcal{S}_2$) where this calculation returns an unsatisfactory $\hat{n}$ is given in Figure 4.1b. We observe that even though the $\hat{n}$ calculated for $\mathcal{S}_2$ does not always result in highly accurate learned frequency information, it does match the decay of the Hankel singular values (Figure 4.1d). For a discussion of when this occurs, see Section 4.6.



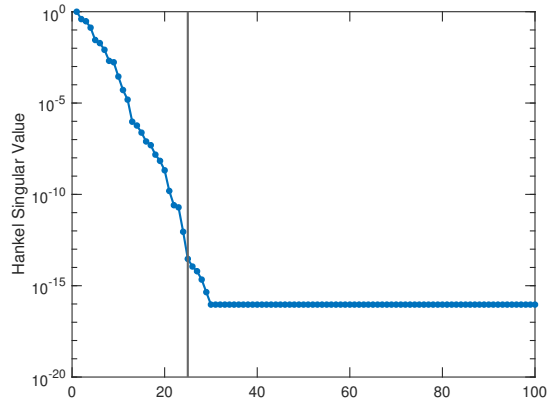
(a) Decay of max and mean relative error in learned transfer function values for $\mathcal{S}_1$.



(b) Decay of max and mean relative error in learned transfer function values for $\mathcal{S}_2$.



(c) Decay of normalized Hankel singular values for $\mathcal{S}_1$.



(d) Decay of normalized Hankel singular values for $\mathcal{S}_2$.

Figure 4.1: Decay of relative error in estimation of $M_0^{\hat{n}}$ approximation as $\hat{n} \to n$ and decay of normalized Hankel singular values. The red lines in (a) and (b) indicate where the requirements of Theorem 3.7 were not met. The vertical black lines in (a) and (b) indicate the $\hat{n}$ calculated using Theorem 4.1, and the vertical black lines in (c) and (d) indicate when the normalized Hankel singular values fall below 10^{-13}.

4.2 Windowing for Multiple Estimates

Lemma 3.6 led to (3.24), a theoretical method to calculate M_0 directly from the data $\mathbb{U} \in \mathbb{R}^{T+1}$ and $\mathbb{Y} \in \mathbb{R}^{T+1}$. Section 4.1 gave us a way to calculate an approximate order $\hat{n}$ to replace n in (3.24), extending the data informativity framework to be data driven. Now

define and consider the matrix

$$\mathbf{G}_{\hat{n}} \coloneqq \begin{bmatrix} \mathbb{H}_{\hat{n}}(\mathbb{U}) \\ \mathbb{H}_{\hat{n}}(\mathbb{Y}) \end{bmatrix} \in \mathbb{C}^{2(\hat{n}+1)\times(T-\hat{n}+1)}. \tag{4.3}$$

If $T > 3\hat{n}+1$, then $\mathbf{G}_{\hat{n}}$ has more columns than rows. Note that there is no restriction on how large T is. In fact we would prefer T to be large ($T \gg 3\hat{n}+1$), so that we have more data to work with. However, $T \gg 3\hat{n}+1$ leads to $\mathbf{G}_{\hat{n}}$ (possibly only numerically) having full row rank for almost all systems, due to the overabundance of columns. This poses a problem for our algorithm, since to satisfy (3.26) of Theorem 3.7, we require $\mathbf{G}_n$ to have row rank $p < 2(\hat{n}+1)$, i.e., $\mathbf{G}_n$ cannot have full row rank. If the row rank of $\mathbf{G}_{\hat{n}}$ is greater than or equal to $2(\hat{n}+1)$, we are unable to append a vector and increase the rank of $\mathbf{G}_{\hat{n}}$, as required by (3.26).

To address this issue, we choose some $t \approx 3\hat{n} < T$ and break our data into $T-t+1$ windows of length $t+1$

$$\mathbb{U}_k = \begin{bmatrix} u_k & \dots & u_{k+t} \end{bmatrix} \in \mathbb{R}^{t+1}, \quad \mathbb{Y}_k = \begin{bmatrix} y_k & \dots & y_{k+t} \end{bmatrix} \in \mathbb{R}^{t+1}, \quad k = 0, \dots, T-t. \tag{4.4}$$

We can now define

$$\mathbf{G}_{k,\hat{n}} \coloneqq \begin{bmatrix} \mathbb{H}_{\hat{n}}(\mathbb{U}_k) \\ \mathbb{H}_{\hat{n}}(\mathbb{Y}_k) \end{bmatrix} \in \mathbb{C}^{2(\hat{n}+1)\times(t-\hat{n}+1)} \tag{4.5}$$

to be the matrix (4.3) constructed using only the data in $\mathbb{U}_k$ and $\mathbb{Y}_k$ and with the calculated $\hat{n}$ in place of the true n. This provides us with two advantages, the first being that we can tune t so that $\operatorname{span} \mathbf{G}_{k,\hat{n}} \neq \mathbb{C}^{2(\hat{n}+1)}$, which allows us to satisfy (3.26) of Theorem 3.7. The second advantage is that for any given $\sigma \in \mathbb{C}$ we now have multiple estimates $M_{0,k}$ for $H(\sigma)$, which gives us an approximation to the error in the calculation.

We solve

$$\begin{bmatrix} & 0 \\ \mathbf{G}_{k,\hat{n}} & \\ & -\gamma_{\hat{n}}(\sigma) \end{bmatrix} \begin{bmatrix} \xi \\ M_{0,k} \end{bmatrix} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix} \tag{4.6}$$

for each $k = 0, \ldots, T-t$ to obtain $\{M_{0,k}\}_{k=0}^{T-t}$, $T-t+1$ estimates of $H(\sigma)$. We then take our estimate to $H(\sigma)$ to be the mean of our estimates, i.e.,

$$M_0 \coloneqq \frac{1}{T-t} \sum_{k=0}^{T-t+1} M_{0,k}. \tag{4.7}$$

In our numerical experiments, we observe that the standard deviation normalized by M_0 of the estimates $\{M_{0,k}\}_{k=0}^{T-t}$ provides an approximation to the relative error

$$\epsilon = \left| \frac{H(\sigma) - M_0}{H(\sigma)} \right|. \tag{4.8}$$

Figure 4.2 plots the normalized standard deviation vs. the relative error in calculation of M_0 for the system $\mathcal{S}_1$ from Section 4.1. We observe that the normalized standard deviation of the estimates $M_{0,k}$ typically well approximates the relative error in the learned transfer function value (4.8). This is a valuable addition to our method, since we now have an approximation to the relative error in learned transfer function value $M_0(\sigma)$, even when we do not know the true value of $H(\sigma)$.

4.3 Improving Conditioning

In this section we present two ways to decrease the condition number of the linear system (4.6). The first is to replace $\mathbf{G}_{k,\hat{n}}$ with an orthogonal subspace, and the second is to normalize the vector $[0^\top, -\gamma_{\hat{n}}(\sigma)^\top]^\top$. Proofs of the equivalence of each update to the original problem (4.6) are provided. We also provide a faster, more robust method for checking condition

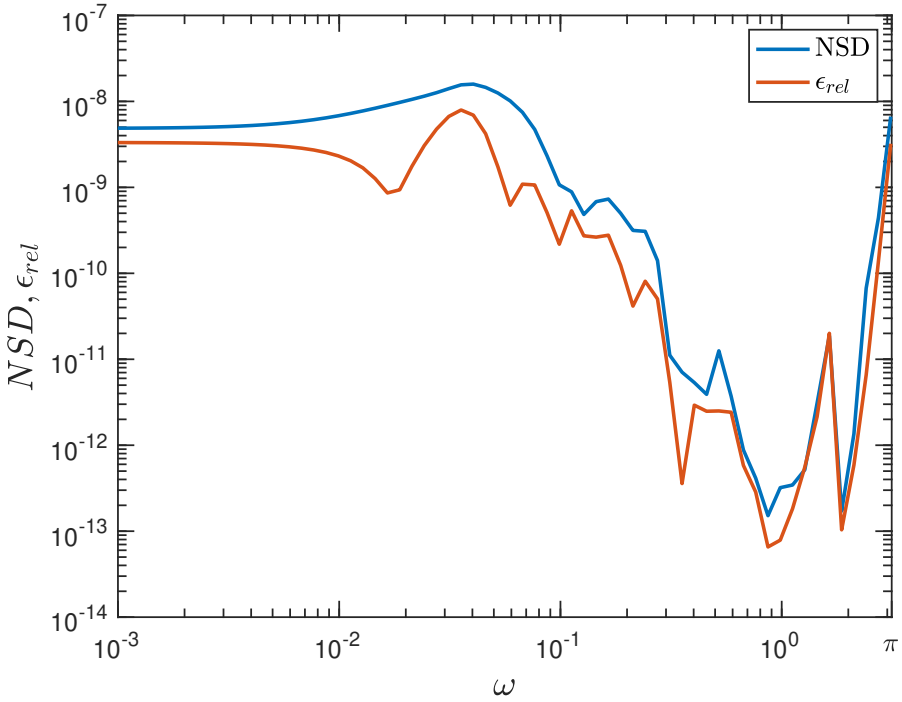


Figure 4.2: The normalized standard deviation (NSD) of $\{M_{0,k}(e^{\mathbf{i}\omega})\}_{k=0}^{T-t}$ provides a good estimator for the relative error (ϵ_{rel}) in $M_0(e^{\mathbf{i}\omega})$

(3.26) of Theorem 3.7. Finally, we provide an explicit formula for the condition number of any subunitary matrix appended with an extra column, and prove that normalizing this appended column minimizes the condition number of the resulting matrix over all possible scalings of the appended column.

4.3.1 Orthogonal Subspace

Section 4.2 gave a practical method to construct linear systems as in (4.6) that satisfy the existence and uniqueness conditions (3.25) and (3.26). However, linear systems of the form (4.6) contain Hankel matrices, which are typically ill-conditioned [1, 6]. Proposition 4.2 provides a way to generate linear systems that typically have lower condition numbers than (4.6), while maintaining the same theoretical solution for $M_{0,k}$.

Proposition 4.2. *Let*

$$\begin{gathered}\mathbf{U}_k\Sigma\mathbf{V}^H = \mathbf{G}_{k,\hat{n}},\\ \mathbf{U}_k \in \mathbb{R}^{2(\hat{n}+1)\times p}, \quad \Sigma \in \mathbb{R}^{p\times p}, \quad \mathbf{V}^H \in \mathbb{R}^{p\times(t-\hat{n}+1)}\end{gathered} \tag{4.9}$$

for some $p < t-\hat{n}+1$ be the SVD of $\mathbf{G}_{k,\hat{n}}$. Let $\sigma \in \mathbb{C}$. Then solving

$$\begin{bmatrix}\mathbf{G}_{k,\hat{n}} & 0\\ & -\gamma_{\hat{n}}(\sigma)\end{bmatrix}\begin{bmatrix}\xi\\ M_{0,k}\end{bmatrix} = \begin{bmatrix}\gamma_{\hat{n}}(\sigma)\\ 0\end{bmatrix} \tag{4.10}$$

and solving

$$\begin{bmatrix}\mathbf{U}_k & 0\\ & -\gamma_{\hat{n}}(\sigma)\end{bmatrix}\begin{bmatrix}\hat{\xi}\\ M_{0,k}\end{bmatrix} = \begin{bmatrix}\gamma_{\hat{n}}(\sigma)\\ 0\end{bmatrix}, \tag{4.11}$$

where $\xi \in \mathbb{C}^{t-\hat{n}}$ and $\hat{\xi} \in \mathbb{C}^{p}$ give the same solution for $M_{0,k}$.

Proof. Let $[\ \xi^T\ M_{0,k}]^T$ solve (4.10). Then inserting the SVD of $\mathbf{G}_{k,\hat{n}}$ (4.9) into (4.10), we have

$$\begin{bmatrix}\mathbf{U}_k\Sigma\mathbf{V}^H & 0\\ & -\gamma_{\hat{n}}(\sigma)\end{bmatrix}\begin{bmatrix}\xi\\ M_{0,k}\end{bmatrix} = \begin{bmatrix}\gamma_{\hat{n}}(\sigma)\\ 0\end{bmatrix}, \tag{4.12}$$

and pulling the $\Sigma\mathbf{V}^H$ out of $\mathbf{G}_{k,\hat{n}} = \mathbf{U}_k\Sigma\mathbf{V}^H$ in (4.12) gives

$$\begin{bmatrix}\mathbf{U}_k & 0\\ & -\gamma_{\hat{n}}(\sigma)\end{bmatrix}\begin{bmatrix}\Sigma\mathbf{V}^H & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\xi\\ M_{0,k}\end{bmatrix} = \begin{bmatrix}\gamma_{\hat{n}}(\sigma)\\ 0\end{bmatrix}. \tag{4.13}$$

Finally, rearranging (4.13) gives

$$\begin{bmatrix}\mathbf{U}_k & 0\\ & -\gamma_{\hat{n}}(\sigma)\end{bmatrix}\begin{bmatrix}\Sigma\mathbf{V}^H\xi\\ M_{0,k}\end{bmatrix} = \begin{bmatrix}\gamma_{\hat{n}}(\sigma)\\ 0\end{bmatrix}. \tag{4.14}$$

Thus $M_{0,k}$ is not affected by replacing $\mathbf{G}_{k,\hat{n}}$ with $\mathbf{U}_k$, i.e., (4.11) and (4.10) give the same solution for $M_{0,k}$. □

We note the difference between Proposition 4.2 and simply using the SVD to solve (4.6). Assuming Theorem 3.7 is satisfied, use the SVD to decompose

$$\hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^H = \begin{bmatrix} \mathbf{G}_{k,\hat{n}} & 0 \\ & -\gamma_{\hat{n}}(\sigma) \end{bmatrix} \tag{4.15}$$
$$\hat{\mathbf{U}} \in \mathbb{C}^{2(\hat{n}+1)\times(p+1)}, \hat{\Sigma} \in \mathbb{C}^{(p+1)\times(p+1)}, \hat{\mathbf{V}}^H \in \mathbb{C}^{(p+1)\times(t-\hat{n}+1)},$$

where p is the rank of $\mathbf{G}_{k,\hat{n}}$. Then we solve (4.6) by

$$\begin{bmatrix} \xi \\ M_{0,k} \end{bmatrix} = \hat{\mathbf{V}}\hat{\Sigma}^{-1}\hat{\mathbf{U}}^H \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix}.$$

This method of solving the linear system (4.6) does not improve the condition number, as the ill-conditioning in

$$\begin{bmatrix} \mathbf{G}_{k,\hat{n}} & 0 \\ & -\gamma_{\hat{n}}(\sigma) \end{bmatrix}$$

is represented as ill conditioning in $\hat{\Sigma}$. In contrast, Proposition 4.2 does not require us to keep Σ. This allows us to compute a much better conditioned basis for the range of $\mathbf{G}_{k,\hat{n}}$ while not keeping the ill-conditioning of $\mathbf{G}_{k,\hat{n}}$. For references on linear system solving and conditioning of linear systems, see [12, 27].

Now, for every window k, we first find an orthonormal basis $\mathbf{U}_k$ for $\mathbf{G}_{k,\hat{n}}$, then compute $M_{0,k}$ by solving

$$\begin{bmatrix} \mathbf{U}_k & 0 \\ & -\gamma_{\hat{n}}(\sigma) \end{bmatrix} \begin{bmatrix} \xi \\ M_{0,k} \end{bmatrix} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix}. \tag{4.16}$$

Note that we still refer to all but the last component of the solution vector as ξ. This is because these values play no role in the calculation of M_0.

There are more advantages to using $\mathbf{U}_k$ in place of $\mathbf{G}_{k,\hat{n}}$ than just a reduction in condition number. The first is that the size of the linear system (4.16) is $2(\hat{n}+1) \times (p+1)$, while the size of the linear system (4.6) is $2(\hat{n}+1) \times (t-\hat{n}+1)$. Since $p \leq t-\hat{n}$, (4.16) is a smaller problem. While we do still have to compute an SVD to find $\mathbf{U}_k$, these orthonormal bases can be precomputed, which leads to an improvement when the number of points to learn $H(z)$ becomes large. Second, the matrix

$$\begin{bmatrix} & 0 \\ \mathbf{U}_k & \\ & -\gamma_{\hat{n}}(\sigma) \end{bmatrix}$$

is only one step of Gram-Schmidt orthogonalization away from being a subunitary matrix, so QR decompositions can be computed efficiently. While we do still have to compute $\mathbf{U}_k$ for each k, these can be precomputed, decreasing the computational complexity of the algorithm.

The final advantage of using $\mathbf{U}_k$ in place of $\mathbf{G}_{k,\hat{n}}$ is that we no longer have to calculate the rank condition (3.26). Instead, we calculate

$$\left\| (\mathbf{I} - \mathbf{U}\mathbf{U}^H) \begin{bmatrix} 0 \\ \\ -\gamma_{\hat{n}}(\sigma) \end{bmatrix} \right\|. \tag{4.17}$$

If the value of (4.17) is nonzero, then appending the vector $[0^\top - \gamma_{\hat{n}}(\sigma)^\top]^\top$ to the matrix $\mathbf{U}_k$ causes

$$\operatorname{rank} \begin{bmatrix} & 0 \\ \mathbf{U}_k & \\ & -\gamma_{\hat{n}}(\sigma) \end{bmatrix} = \operatorname{rank} \mathbf{U}_k + 1, \tag{4.18}$$

which shows condition (3.26) is satisfied. This change marks a significant improvement to

our algorithm, since numerical rank computations can at times be ambiguous. We also gain an advantage in computational complexity, since the main cost of (4.17) is two matrix-vector products ($\mathcal{O}((\hat{n}+1)(p+1))$) while the main cost of (3.26) is a rank revealing factorization (QR or SVD), which is $\mathcal{O}((\hat{n}+1)^2(t-\hat{n}+1))$.

4.3.2 Normalization

In this section, we show that we are able to scale the last column of

$$\begin{bmatrix} & 0 \\ \mathbf{U}_k & \\ & -\gamma_{\hat{n}}(\sigma) \end{bmatrix} \coloneqq \begin{bmatrix} \mathbf{U}_k & \mathbf{z} \end{bmatrix}$$

by $1/\|\mathbf{z}\|$. This will further reduce the condition number of (4.16). Normalizing $\mathbf{z}$ to reduce the condition number of (4.16) is motivated by Theorem 3.5b of [40], presented in Theorem 4.3.

Theorem 4.3. *(van der Sluis, [40]) For any matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$, if $\mathbf{D}$ is a diagonal matrix for which $\mathbf{AD}$ has normalized columns in the 2-norm, then*

$$1 \leq \frac{\kappa_2(\mathbf{AD})}{\kappa_2(\mathbf{A}\overline{\mathbf{D}})} \leq \sqrt{n}, \tag{4.19}$$

where

$$\overline{\mathbf{D}} = \underset{\left\{\substack{\Delta\in\mathbb{C}^{m\times m} \text{ and} \\ \Delta \text{ is diagonal}}\right\}}{\operatorname{argmin}} \kappa_2(\mathbf{A}\Delta) \tag{4.20}$$

Note that the columns of $\mathbf{U}_k$ are orthonormal, so they already have unit norm. Additionally, since $\mathbf{U}_k$ being orthonormal is helpful for quickly finding a QR factorization of $[\ \mathbf{U}_k\ \mathbf{z}]$, we do not want to scale any columns of $\mathbf{U}_k$. Thus, we will consider only diagonal scaling matrices

$\mathbf{D}$ of the form

$$\mathbf{D} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \delta \end{bmatrix}. \tag{4.21}$$

Note that post multiplication of $[\ \mathbf{U}_k\ \mathbf{z}]$ by matrices of the form (4.21) is equivalent to scaling $\mathbf{z}$ by δ.

Theorem 4.3 guarantees that if we scale $\mathbf{z}$ in $[\ \mathbf{U}_k\ \mathbf{z}]$ to be normalized in the 2-norm, then we are at most a factor of $\sqrt{p+1}$ away from the optimal diagonal scaling. In Section 4.3.3, we will prove that normalizing $\mathbf{z}$ is the optimal scaling. We now state and prove Proposition 4.4, which allows us to scale $\mathbf{z}$ by any non zero number and still recover $M_{0,k}$

Proposition 4.4. *Let $\delta \neq 0 \in \mathbb{R}$. If $\hat{\mathbf{z}} = \delta\mathbf{z}$ and $[\ \xi^\top\ M_{0,k}]^\top$ solves (4.16), then*

$$\begin{bmatrix} \mathbf{U}_k & \hat{\mathbf{z}} \end{bmatrix} \begin{bmatrix} \xi \\ \frac{1}{\delta} M_{0,k} \end{bmatrix} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix}. \tag{4.22}$$

Proof. We note that for any matrix $\mathbf{M} \in \mathbb{C}^{m\times n}$, any vector $\mathbf{b} \in \mathbb{C}^m$ and any invertible diagonal matrix $\mathbf{D} \in \mathbb{C}^{n\times n}$,

$$\mathbf{M}\mathbf{x} = \mathbf{b} \iff \mathbf{M}\mathbf{D}\mathbf{D}^{-1}\mathbf{x} = \mathbf{b}.$$

So we can solve

$$\mathbf{M}\mathbf{D}\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = \mathbf{D}^{-1}\mathbf{y}.$$

A more detailed proof follows. Let $[\ \xi^\top\ M_{0,k}]^\top$ solve (4.16), i.e.,

$$\begin{bmatrix} \mathbf{U}_k & \mathbf{z} \end{bmatrix} \begin{bmatrix} \xi \\ M_{0,k} \end{bmatrix} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix}. \tag{4.23}$$

Then expanding the left hand side of (4.23) leads to

$$\mathbf{U}_k\xi + M_{0,k}\mathbf{z} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix}, \tag{4.24}$$

and multiplying M_0 by $\frac{\delta}{\delta}$ gives

$$\mathbf{U}_k\xi + M_{0,k}\frac{1}{\delta}\delta\mathbf{z} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix}. \tag{4.25}$$

Finally, we rewrite (4.25) as a matrix-vector product to obtain

$$\begin{bmatrix} \mathbf{U}_k & \hat{\mathbf{z}} \end{bmatrix} \begin{bmatrix} \xi \\ \frac{1}{\delta}M_{0,k} \end{bmatrix} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix}. \tag{4.26}$$

□

So, taking $\delta = \frac{1}{\|\mathbf{z}\|}$, we are now able to solve

$$\begin{bmatrix} \mathbf{U}_k & \frac{\mathbf{z}}{\|\mathbf{z}\|} \end{bmatrix} \begin{bmatrix} \xi \\ \hat{M}_{0,k} \end{bmatrix} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix} \tag{4.27}$$

and set

$$M_{0,k} = \|\mathbf{z}\|\hat{M}_{0,k} \tag{4.28}$$

to obtain an estimate to the transfer function at σ by solving a (hopefully) better conditioned problem. Section 4.3.3 extends these results, providing a proof that normalizing $\mathbf{z}$ is indeed optimal.

4.3.3 Explicit Condition Number Formula

We now present an explicit formula (Proposition 4.6) for the condition number of any subunitary matrix $\mathbf{U} \in \mathbb{C}^{m\times n}$ appended with an additional column $\mathbf{z} \in \mathbb{C}^m$. We will also prove that normalizing $\|\mathbf{z}\|$ (i.e., choosing $\delta = \frac{1}{\|\mathbf{z}\|}$, $\hat{\mathbf{z}} = \delta\mathbf{z}$ in (4.26)) minimizes the condition number

$$\kappa_2\left(\begin{bmatrix}\mathbf{U} & \delta\mathbf{z}\end{bmatrix}\right)$$

over all numbers $\delta \in \mathbb{R}$. Our strategy will be to analyze the eigenvalues of $[\ \mathbf{U}\ \mathbf{z}][\ \mathbf{U}\ \mathbf{z}]^H = \mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$, and investigate how they change when $\|\mathbf{z}\|$ is varied.

We first present Lemma 4.5 which tells us where two eigenvectors of $\mathbf{U}\mathbf{U}^H+\mathbf{z}\mathbf{z}^H$ with nonzero eigenvalue lie.

Lemma 4.5. *Let $\mathbf{U} \in \mathbb{C}^{m\times n}, m > n$ be subunitary. Let $\mathbf{z} \in \mathbb{C}^m$. Define*

$$\mathbf{u} = \mathbf{U}\mathbf{U}^H\mathbf{z} \quad \textit{and} \quad \mathbf{v} = (\mathbf{I} - \mathbf{U}\mathbf{U}^H)\mathbf{z}$$

so that $\mathbf{u}+\mathbf{v} = \mathbf{z}$. Assume $\mathbf{v} \neq 0$ and $\mathbf{u} \neq 0$. Then there are two eigenvectors of $\mathbf{U}\mathbf{U}^H+\mathbf{z}\mathbf{z}^H$, $\mathbf{x}_r$ and $\mathbf{x}_s$, with eigenvalues $\lambda_r \neq 0$ and $\lambda_s \neq 0$ such that

$$\mathbf{x}_r \in \operatorname{span}\{\mathbf{u}, \mathbf{v}\}$$

and

$$\mathbf{x}_s \in \operatorname{span}\{\mathbf{u}, \mathbf{v}\}.$$

Further, if $\mathbf{x}_i \notin \operatorname{span}\{\mathbf{u}, \mathbf{v}\}$ is an eigenvector of $\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$ with eigenvalue λ_i, then

$$\lambda_i = 1 \quad \textit{or} \quad \lambda_i = 0.$$

Proof. Since $\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$ is Hermitian, it is unitarily diagnalizable as

$$\mathbf{X}\Lambda\mathbf{X}^H.$$

Since $\mathbf{v} \neq 0$ and $\mathbf{U}$ is subunitary (hence full column rank), there are $n+1$ nonzero eigenvalues of $\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$. Also since $\mathbf{U}$ has full column rank, we can find $n-1$ vectors $\mathbf{u}_i \in \text{Range}(\mathbf{U})$ such that

$$\mathbf{u}_i^H \mathbf{u} = 0, \quad i = 1, 2, \ldots, n-1.$$

Since each $\mathbf{u}_i \in \text{Range}(\mathbf{U})$, we also have

$$\mathbf{u}_i^H \mathbf{v} = 0, \quad i = 1, 2, \ldots, n-1.$$

Note that each $\mathbf{u}_i$ is an eigenvector of $\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$ with eigenvalue $\lambda_i = 1$

$$(\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H)\mathbf{u}_i = (\mathbf{U}\mathbf{U}^H + (\mathbf{u}+\mathbf{v})(\mathbf{u}+\mathbf{v})^H)\mathbf{u}_i = \mathbf{u}_i.$$

Now note that since $\mathbf{u} \neq 0$ and $\mathbf{v} \neq 0$, neither $\mathbf{u}$ nor $\mathbf{v}$ is an eigenvector of $\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$:

$$(\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H)\mathbf{u} = (\mathbf{U}\mathbf{U}^H + (\mathbf{u}+\mathbf{v})(\mathbf{u}+\mathbf{v})^H)\mathbf{u} = \mathbf{u} + \|\mathbf{u}\|^2(\mathbf{u}+\mathbf{v}), \tag{4.29}$$

$$(\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H)\mathbf{v} = (\mathbf{U}\mathbf{U}^H + (\mathbf{u}+\mathbf{v})(\mathbf{u}+\mathbf{v})^H)\mathbf{v} = \|\mathbf{v}\|^2(\mathbf{u}+\mathbf{v}), \tag{4.30}$$

but both $\mathbf{u}$ and $\mathbf{v}$ are in $\text{Range}(\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H)$:

$$(\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H)\left(\frac{\mathbf{u}}{\|\mathbf{u}\|^2} - \frac{\mathbf{v}}{\|\mathbf{v}\|^2}\right) = (\mathbf{U}\mathbf{U}^H + (\mathbf{u}+\mathbf{v})(\mathbf{u}+\mathbf{v})^H)\left(\frac{\mathbf{u}}{\|\mathbf{u}\|^2} - \frac{\mathbf{v}}{\|\mathbf{v}\|^2}\right) = \frac{\mathbf{u}}{\|\mathbf{u}\|^2}, \tag{4.31}$$

and (4.30) and (4.31) show $\mathbf{v} \in \text{Range}(\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H)$. Recall that we have already found

$n-1$ eigenvectors $\mathbf{x}_i$ with eigenvalue $\lambda_i = 1$, and we have a total of $n+1$ eigenvectors with nonzero eigenvalue. Call the remaining two eigenvectors with nonzero eigenvalue $\mathbf{x}_r$ and $\mathbf{x}_s$. Since the $n-1$ eigenvectors of $\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$ with eigenvalue 1 are orthogonal to $\mathbf{u}$ and $\mathbf{v}$, and both $\mathbf{u} \in \text{Range}(\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H)$ and $\mathbf{v} \in \text{Range}(\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H)$ while not being eigenvectors, they both must be in the span of $\mathbf{x}_r$ and $\mathbf{x}_s$. So since $\mathbf{x}_s$ and $\mathbf{x}_r$ are linearly independent,

$$\text{span}\{\mathbf{u}, \mathbf{v}\} = \text{span}\{\mathbf{x}_r, \mathbf{x}_s\},$$

so

$$\mathbf{x}_r \in \text{span}\{\mathbf{u}, \mathbf{v}\}$$

and

$$\mathbf{x}_s \in \text{span}\{\mathbf{u}, \mathbf{v}\}.$$

□

Proposition 4.6. *Let* $\mathbf{U} \in \mathbb{C}^{m\times n}$ *with* $n < m$ *be subunitary. Let* $\mathbf{z} \in \mathbb{C}^m$, $\mathbf{u} = \mathbf{U}\mathbf{U}^H\mathbf{z}$, $\mathbf{v} = (\mathbf{I} - \mathbf{U}\mathbf{U}^H)\mathbf{z}$, *and* $\nu = \|\mathbf{z}\|$. *Assume* $\|\mathbf{v}\| \neq 0$ *and* $\|\mathbf{u}\| \neq 0$. *Then the matrix* $[\ \mathbf{U}\ \mathbf{z}]$ *has condition number*

$$\kappa_2\left([\ \mathbf{U}\ \mathbf{z}]\right) = \frac{\sqrt{1+\nu^2+\sqrt{1+\nu^4+2\nu^2-4\|\mathbf{v}\|^2}}}{\sqrt{1+\nu^2-\sqrt{1+\nu^4+2\nu^2-4\|\mathbf{v}\|^2}}}. \tag{4.32}$$

Proof. We will consider the eigenvalues of $[\ \mathbf{U}\ \mathbf{z}][\ \mathbf{U}\ \mathbf{z}]^H = \mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$. First we expand

$$\begin{aligned} \mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H &= \mathbf{U}\mathbf{U}^H + (\mathbf{u}+\mathbf{v})(\mathbf{u}+\mathbf{v})^H \\ &= \mathbf{U}\mathbf{U}^H + \mathbf{u}\mathbf{u}^H + \mathbf{v}\mathbf{v}^H + \mathbf{u}\mathbf{v}^H + \mathbf{v}\mathbf{u}^H. \end{aligned} \tag{4.33}$$

By Lemma 4.5, there are two eigenvectors of $\mathbf{U}\mathbf{U}^H + \mathbf{z}\mathbf{z}^H$ in $\text{span}\{\mathbf{u}, \mathbf{v}\}$. So for $\alpha_i \in \mathbb{C}$ and

$\beta_i \in \mathbb{C}$, and $\lambda_i \in \mathbb{R}$, for $i = 1, 2$, we have

$$(\mathbf{U}\mathbf{U}^H + \mathbf{u}\mathbf{u}^H + \mathbf{v}\mathbf{v}^H + \mathbf{u}\mathbf{v}^H + \mathbf{v}\mathbf{u}^H)(\alpha_i\mathbf{u} + \beta_i\mathbf{v}) = \lambda_i(\alpha_i\mathbf{u} + \beta_i\mathbf{v}), \tag{4.34}$$

i.e, $(\alpha_i\mathbf{u} + \beta_i\mathbf{v}, \lambda_i)$ are eigenpairs for $i = 1, 2$. Note that since $\mathbf{u}$ and $\mathbf{v}$ are not eigenvectors, neither α_i nor β_i are 0. Now expanding (4.34) (and dropping subscripts), we obtain

$$\alpha\mathbf{u} + \alpha\|\mathbf{u}\|^2\mathbf{u} + \alpha\|\mathbf{u}\|^2\mathbf{v} + \beta\|\mathbf{v}\|^2\mathbf{v} + \beta\|\mathbf{v}\|^2\mathbf{u} = \lambda(\alpha\mathbf{u} + \beta\mathbf{v}), \tag{4.35}$$

and grouping the terms of (4.35) yields

$$((1 - \lambda)\alpha + \alpha\|\mathbf{u}\|^2 + \beta\|\mathbf{v}\|^2)\mathbf{u} + (\alpha\|\mathbf{u}\|^2 + \beta\|\mathbf{v}\|^2 - \lambda\beta)\mathbf{v} = 0. \tag{4.36}$$

Now, since $\mathbf{u}$ and $\mathbf{v}$ are orthogonal, their linear combination is 0 if and only if both coefficients are 0. So we obtain the system of equations

$$\begin{aligned} (1 - \lambda)\alpha + \alpha\|\mathbf{u}\|^2 + \beta\|\mathbf{v}\|^2 &= 0 \\ \alpha\|\mathbf{u}\|^2 + \beta\|\mathbf{v}\|^2 - \lambda\beta &= 0. \end{aligned} \tag{4.37}$$

Since an eigenvector is still an eigenvector regardless of its norm, we are free to arbitrarily choose α or β. Setting $\beta = 1$ and solving (4.37) for λ and α yields

$$\begin{aligned} \alpha &= \frac{1}{2\|\mathbf{u}\|^2}\left(1 - \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 \pm \sqrt{(\|\mathbf{v}\|^2 - \|\mathbf{u}\|^2 - 1)^2 + 4\|\mathbf{u}\|^2\|\mathbf{v}\|^2}\right) \\ \lambda &= \frac{1}{2}\left(1 - \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 \pm \sqrt{(1 - \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2)^2 + 4\|\mathbf{u}\|^2\|\mathbf{v}\|^2}\right) + \|\mathbf{v}\|^2 \end{aligned} \tag{4.38}$$

Recalling that $\nu^2 = \|\mathbf{z}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2$, simplifying (4.38) leads to

$$\lambda = \frac{1}{2}\left(1 + \nu^2 \pm \sqrt{1 + \nu^4 + 2\nu^2 - 4\|\mathbf{v}\|^2}\right). \tag{4.39}$$

Finally, Lemma 4.5 also guarantees that all eigenvectors not in span$\{\mathbf{u}, \mathbf{v}\}$ have eigenvalue 1 or 0, so the eigenvalues found in (4.39) must be the first and $(n+1)$-st. Thus

$$\sqrt{\frac{\lambda_1}{\lambda_{n+1}}} = \kappa_2\left([\ \mathbf{U}\ \mathbf{z}]\right) = \frac{\sqrt{1+\nu^2+\sqrt{1+\nu^4+2\nu^2-4\|\mathbf{v}\|^2}}}{\sqrt{1+\nu^2-\sqrt{1+\nu^4+2\nu^2-4\|\mathbf{v}\|^2}}} \tag{4.40}$$

as claimed. □

Note that as $\|\mathbf{v}\| \to 0$, the formula (4.32) approaches infinity, which reflects the matrix $[\ \mathbf{U}\ \mathbf{z}]$ approaching rank deficiency. In the case where $\|\mathbf{u}\| \to 0$, i.e., $\|\mathbf{v}\| \to \nu$, we have

$$\sqrt{1+\nu^4+2\nu^2-4\|\mathbf{v}\|^2} \to |\nu^2-1|,$$

and the quotient

$$\kappa_2\left([\ \mathbf{U}\ \mathbf{z}]\right) \to \frac{\sqrt{1+\nu^2+|\nu^2-1|}}{\sqrt{1+\nu^2-|\nu^2-1|}}, \tag{4.41}$$

which is minimized when $\nu = 1$ since $\kappa_2([\ \mathbf{U}\ \mathbf{z}]) \geq 1$. There is no other value of ν that could minimize (4.41) since $|\nu^2-1| > 0$ for all values of ν except 1. Since we subtract $|\nu^2-1|$ from the denominator and add it to the numerator in (4.41), $|\nu^2-1| = 0$ is clearly required for optimality. This result can also be seen by the fact that $[\mathbf{U}\ \mathbf{z}]$ is a subunitary or unitary matrix when $(\mathbf{I}-\mathbf{U}\mathbf{U}^H)\mathbf{z} = \mathbf{z}$ and $\|\mathbf{z}\| = 1$. So we have shown that for the special case where $\|\mathbf{v}\| = \|\mathbf{z}\| = \nu$, normalizing $\mathbf{z}$ is indeed optimal.

Consider a fixed $\mathbf{z}$. Then, independent of $\|\mathbf{z}\|$, there exists $0 < \eta < 1$ such that $\|\mathbf{v}\| = \eta\|\mathbf{z}\| = \eta\nu$. Substituting $\|\mathbf{v}\| = \eta\nu$ in (4.40) and squaring gives

$$\kappa_2^2\left([\ \mathbf{U}\ \mathbf{z}]\right) = K(\nu,\eta) = \frac{1+\nu^2+\sqrt{1+\nu^4+2\nu^2-4\eta^2\nu^2}}{1+\nu^2-\sqrt{1+\nu^4+2\nu^2-4\eta^2\nu^2}}. \tag{4.42}$$

Then differentiating with respect to ν gives

$$\frac{\partial K}{\partial \nu}(\nu,\eta) = \frac{8\eta^2\nu(\nu^2-1)}{\sqrt{1+(2-4\eta^2)\nu^2+\nu^4}\left(1+\nu^2-\sqrt{1+(2-4\eta^2)\nu^2+\nu^4}\right)^2}, \tag{4.43}$$

which leads to Proposition 4.7.

Proposition 4.7. *Let* $\mathbf{U} \in \mathbb{C}^{m\times n}$ *with* $n < m$ *be subunitary. Let* $\mathbf{z} \in \mathbb{C}^m$, *where* $\mathbf{z} \notin$ Range($\mathbf{U}$). *Then* $\delta = \frac{1}{\|\mathbf{z}\|}$ *minimizes* $\kappa_2([\mathbf{U}\ \delta\mathbf{z}])$ *over all* $\delta \in \mathbb{R}$.

Proof. If $\mathbf{z} \perp$ Range($\mathbf{U}$), then normalizing z leads to $[\mathbf{U}\ \mathbf{z}]$ being subunitary (or potentially unitary). So $\kappa_2([\mathbf{U}\ \mathbf{z}]) = 1$, which is clearly minimized.

If $\mathbf{z} \not\perp$ Range($\mathbf{U}$), then $\kappa_2([\mathbf{U}\ \mathbf{z}])$ is given by (4.40) and, when squared, has derivative (4.43). Note that the denominator of (4.43) is always greater than zero, and $8\eta^2\nu$ is also always greater than zero by our assumptions. So the sign of (4.43) is determined by

$$(\nu^2-1).$$

Since ν^2-1 is 0 if $\nu = 1$, negative if $0 < \nu < 1$, and positive for $\nu > 1$, $\nu = 1$ is the only local minimum of (4.40) for the feasible values of ν and η.

Then since (4.40) is decreasing for all allowed values of ν to the left of 1 and increasing for all $\nu > 1$, $\nu = 1$ is the global minimum. □

We illustrate the result of Proposition 4.7 in Figure 4.3 by plotting $\kappa_2([\mathbf{U}\ \mathbf{z}])$ for different values of ν and $\eta = \|\mathbf{z}\|/\nu$. We observe in Figure 4.3 that for a set $\|\mathbf{v}\|/\nu$, $\nu = 1$ minimizes (4.32) over the range of tested ν values, as expected. These results lead us to normalize $\mathbf{z}$ in

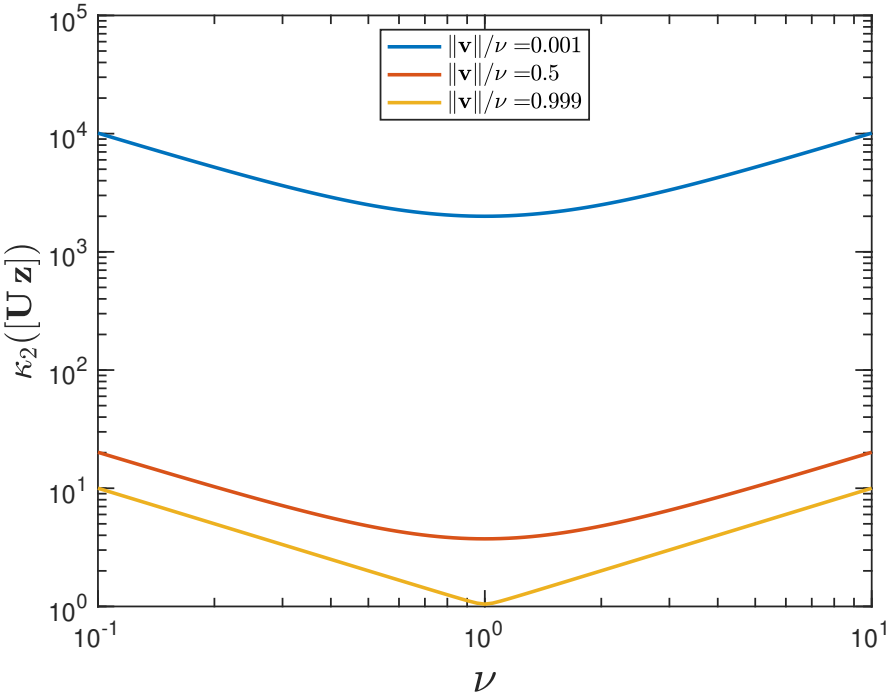


Figure 4.3: $\kappa_2([\mathbf{U}\,\mathbf{z}])$ dependence on ν. For all three $\|\mathbf{v}\|/\nu$ values, $\kappa_2([\mathbf{U}\,\mathbf{z}])$ was minimized when $\nu = 1$.

our algorithm to solve a better conditioned problem

$$\begin{bmatrix} \mathbf{U}_k & \frac{\mathbf{z}}{\|\mathbf{z}\|} \end{bmatrix} \begin{bmatrix} \xi \\ \hat{M}_{0,k} \end{bmatrix} = \begin{bmatrix} \gamma_{\hat{n}}(\sigma) \\ 0 \end{bmatrix}; \quad M_{0,k} = \frac{1}{\|\mathbf{z}\|}\hat{M}_{0,k} \tag{4.44}$$

to calculate $M_{0,k}$.

4.4 Implementation Details

Sections 4.1, 4.2, 4.3.1, and 4.3.2 introduced various numerical improvements to the data informativity framework discussed in Chapter 3. However there are still important implementation details to increase the likelihood of accurate learned frequency domain data

$M_0(\sigma) \approx H(\sigma)$ and to speed up the run time of the algorithm. We will consider some of these details here.

4.4.1 Allowing Least Squares Solutions

The discussion thus far in Chapter 4 has always assumed that we can exactly solve linear systems of the form (4.44). Indeed, this is what condition (3.25) of Theorem 3.7 requires. However, in practice, we can allow least squares solutions to (4.44) with small residual and still get accurate approximations $M_{0,k}(\sigma) \approx H(\sigma)$. We emphasize that all our efforts to decrease computation time and improve conditioning of the problem (4.44), such as Propositions 4.2, 4.4, and 4.6, also apply to the least squares case.

4.4.2 Number of Windows

In Section 4.2, we introduced the idea of windowing the data in $\mathbb{U}$ and $\mathbb{Y}$ to satisfy the requirements of Theorem 3.7. Previously, in (4.7), we compute an estimate $M_{0,k}(\sigma)$ for each of the $T - t + 1$ windows, then accept their average as $M_0(\sigma) \approx H(\sigma)$. In this section, we discuss how many of these $T - t + 1$ windows are needed to obtain an accurate estimate of M_0.

To motivate reducing the number of data windows, we consider the cost of creating the orthogonal subspace for a window. Each orthogonal subspace requires either an SVD or QR decomposition of a matrix $\mathbf{G}_{k,\hat{n}}$ of the form (4.5). These are both $\mathcal{O}(\hat{n}^3)$ algorithms. As assumed in Section 4.2, $T \gg 3\hat{n}$. Computing $T - t + 1$ matrix decompositions of matrices $\mathbf{G}_{k,\hat{n}} \in \mathbb{C}^{2(\hat{n}+1)\times(t-\hat{n}+1)}$ is expensive, and as will be shown in Figure 4.4, largely unnecessary.

To illustrate numerically that we do not need all $T - t + 1$ windows, we choose two points

in the complex plane where we would like to learn M_0 :

$$\sigma_1 = e^{0.1\cdot\mathbf{i}}, \quad \sigma_2 = e^{0.5\cdot\mathbf{i}}.$$

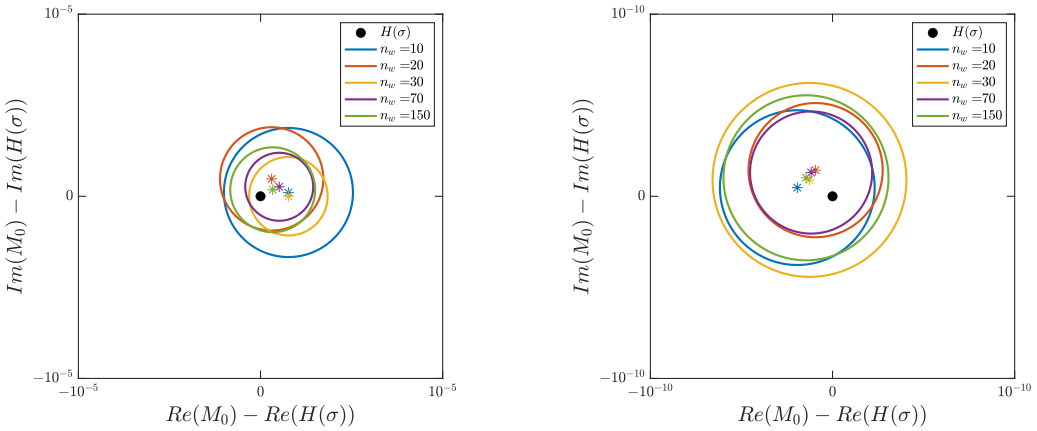
We then calculate an estimate to $M_0(\sigma_i), i = 1, 2$ for the system $\mathcal{S}_2$, first introduced in Section 4.1, using 5 different numbers of windows

$$n_w = 10, 20, 30, 70, 150.$$

The accepted $M_0(\sigma_i)$ is

$$M_0(\sigma_i) = \frac{1}{n_w} \sum_{i=1}^{n_w} M_{0,k}(\sigma_i),$$

where $M_{0,k}(\sigma_i)$ is the value of $M_{0,k}(\sigma_i)$ from (4.44) using the k-th window for σ_i, $i = 1, 2$. We plot these values and their standard deviations in Figure 4.4.



(a) $\sigma_i = \sigma_1$. (b) $\sigma_i = \sigma_2$.

Figure 4.4: Estimated $M_0(\sigma_i)$ for $\mathcal{S}_2$ (stars) and the boundary of the set of points one standard deviation from estimated $M_0(\sigma_i)$ (solid circles) in their estimates for different number of windows n_w.

Observe the scale in Figure 4.4. Figure 4.4a, showing the results for σ_1 where we have relatively low accuracy in learning $M_0(\sigma_1)$ for all values of n_w, shows that each $M_0(\sigma_1)$

deviates from $H(\sigma_1)$ by less than 10^{-5} in both real and imaginary components. Figure 4.4b, showing the results for σ_2 where we have relatively high accuracy in learning $M_0(\sigma_2)$ for all values of n_w, shows that each $M_0(\sigma_2)$ deviates from $H(\sigma_2)$ by less than 10^{-10} in both real and imaginary component. We see in Figure 4.4a, using $n_w = 150$ windows leads to very modest accuracy gains compared to using only $n_w = 10$ windows. In Figure 4.4b, we observe that using $n_w = 30$ windows actually leads to the most accurate approximation to $H(\sigma_2)$. For both cases, for each number of windows n_w, $H(\sigma_i)$ is within one standard deviation of the mean of the estimated $M_{0,k}(\sigma_i)$, further showing that normalized standard deviation in the estimates is a good predictor of the relative error

$$\frac{|M_0(\sigma_i) - H(\sigma_i)|}{|H(\sigma_i)|}.$$

Figure 4.4 leads to the conclusion that using a low number of windows n_w will provide similar accuracy to using a large number of windows. Further, it is reassuring that our standard deviation estimation of the error applies for each choice of n_w tested in Figure 4.4. So if a choice of small n_w leads to a large standard deviation, one can always increase n_w in hopes of better learned transfer function data.

4.4.3 Choice of input

A key choice that led to $n_w = 10 \ll T - t + 1$ windows having nearly the same accuracy as a much higher number of windows in Section 4.4.2 was that we used a Gaussian random noise vector as our input $\mathbb{U}$. Gaussian random noise has (nearly) equal frequency content on every interval of the input. This feature means that the ability of each window to produce good estimates of $M_0(\sigma)$ is fairly uniform for different values of σ.

We note that we have also had success using chirp input signals. Chirp inputs have variable

frequency content throughout the input vector $\mathbb{U}$. For this reason, if we are interested in learning $M_0(\sigma_i)$ for $i = 1, 2, \ldots, m$, where $\{\sigma_i\}_{i=1}^m$ spans a large frequency range, we expect to require a higher number of windows n_w than if our input is Gaussian random noise. This is to ensure that there is a window of the chirp input that can well approximate $M_0(\sigma_i)$. Additionally, if we expect only a subset of our windows to provide quality estimates to $H(\sigma)$, we must take care to only include values of $M_{0,k}(\sigma)$ calculated from this subset of windows in our calculation of accepted $M_0(\sigma)$ and the standard deviation of $M_{0,k}(\sigma)$. Otherwise, if all windows are used, we can expect the accepted $M_0(\sigma)$ to be inaccurate and the standard deviation in $M_{0,k}$ to be large.

For these reasons, in this thesis we recommend using Gaussian random noise as the input vector $\mathbb{U}$ for this algorithm. We also acknowledge that different inputs (or multiple inputs) is an area for future work on this project.

4.5 Algorithm

In this section, we present an algorithmic summary of the results discussed in Chapter 4. At each step, we reference the section or equation that justifies the step. We note that without the work of Chapter 4, this algorithm would simply be to check the conditions of Theorem 3.7 and, if they are met, solve (3.24). The advantages of Algorithm 2 are numerous, and include better conditioning for finding M_0 and an error estimate.

4.6 Impact of Pole Locations

While for most systems, our method is able to learn frequency domain data with good accuracy, some features of the underlying system can cause inaccurate estimation of $H(\sigma)$

Algorithm 2 Numerical Implementation of Data Informativity Framework with Windowing for Multiple Estimates

Require: Time domain input-output data $\mathbb{U}, \mathbb{Y}$ and $\{\sigma_i\}_{i=1}^m \subset \mathbb{C}$, points to learn the transfer function. **return** $\{M_0(\sigma_i)\}_{i=1}^m$, estimates to $H(z)$ at each σ_i

Calculate approximate order $\hat{n}$ (via Theorem 4.1 in Section 4.1).

Set $t \leftarrow k < T$ (we use $k = 3\hat{n}$) (Section 4.2).

Choose number of windows n_w (in most cases, $n_w \approx 10$ will suffice, Section 4.4.2).

Calculate orthogonal bases $\mathbf{U}_k \in \mathbb{C}^{2(\hat{n}+1)\times p}$ for $k = 1, 2, \ldots, n_w$ data windows of length t (Sections 4.2 and 4.3.1).

for $i = 1, 2, \ldots, m$ **do**

Form $\gamma_{\hat{n}}(\sigma_i) \in \mathbb{C}^{\hat{n}+1}$ (See (3.14))

$\mathbf{z} \leftarrow [\ 0^\top\ \gamma_{\hat{n}}(\sigma_i)^\top]^\top \in \mathbb{C}^{2(\hat{n}+1)}$.

for $k = 1, 2, \ldots n_w$ **do**

if $(\mathbf{I} - \mathbf{U}_k\mathbf{U}_k^H)\mathbf{z} \neq 0$ **then**

Calculate (Sections 4.4.1,4.3.2)

$$\begin{bmatrix} \xi \\ \hat{M}_{0,k}(\sigma_i) \end{bmatrix} = \underset{\mathbf{x}\in\mathbb{C}^p}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{U}_k & \frac{\mathbf{z}}{\|\mathbf{z}\|} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \gamma_{\hat{n}}(\sigma_i) \\ 0 \end{bmatrix} \right\|$$

$M_{0,k}(\sigma_i) \leftarrow \frac{1}{\|\mathbf{z}\|}\hat{M}_{0,k}(\sigma_i)$ (from Sections 4.3.2,4.3.3).

end if

end for

$M_0(\sigma_i) \leftarrow \frac{1}{n_w}\sum_{k=1}^{n_w} M_{0,k}$ (from Section 4.2).

end for

at some values of $\sigma \in \mathbb{C}$. These features are clustered poles (Section 4.6.1) and poles that lie strictly on the real axis (Section 4.6.2). The key difference, explored below, is that for systems with clustered poles, the accuracy of the learned data $M_0(\sigma)$ obtained via (4.44) for σ near a pole cluster is low, and the accuracy is independent of $\hat{n}$. For systems with real poles, when $\hat{n}$ obtained via Theorem 4.1 is used to learn $M_0(\sigma)$ for σ near the real axis, we achieve relatively low accuracy. However, when the true system dimension n is used to estimate $M_0(\sigma)$, we attain higher accuracy. A discussion of these features is provided in

Section 4.6.3.

4.6.1 Clustered Poles

The first feature of systems that can make learning $M_0(\sigma)$ difficult is clustering of poles near σ. To illustrate this, we pick four cluster centers

$$z_1 = 0.5, \quad z_2 = 0.7, \quad z_3 = 0.9, \text{ and } \quad z_4 = 0.99.$$

For each cluster center z_i we construct an order $n = 50$ system $\hat{\mathcal{S}}_i$ that has poles clustered within a radius of 0.01 around z_i. So if λ is a pole of $\hat{\mathcal{S}}_i$, then

$$|\lambda - z_i| < 0.01.$$

Each system was constructed with the same random $\mathbf{b} \in \mathbb{R}^n$ and random $\mathbf{c}^\top \in \mathbb{R}^{1\times n}$. For system $\hat{\mathcal{S}}_i$, the matrix $\mathbf{A}_i$ was constructed by

$$\mathbf{A}_i = \mathbf{V}^H \mathbf{M}_i \mathbf{V} \in \mathbb{R}^{n\times n},$$

where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a random orthogonal matrix and $\mathbf{M}_i \in \mathbb{R}^{n\times n}$ is the block diagonal matrix comprised of 2×2 blocks with complex conjugate eigenvalues in the appropriate cluster. So for the system $\hat{\mathcal{S}}_i$ with transfer function $H_i(z)$, we have

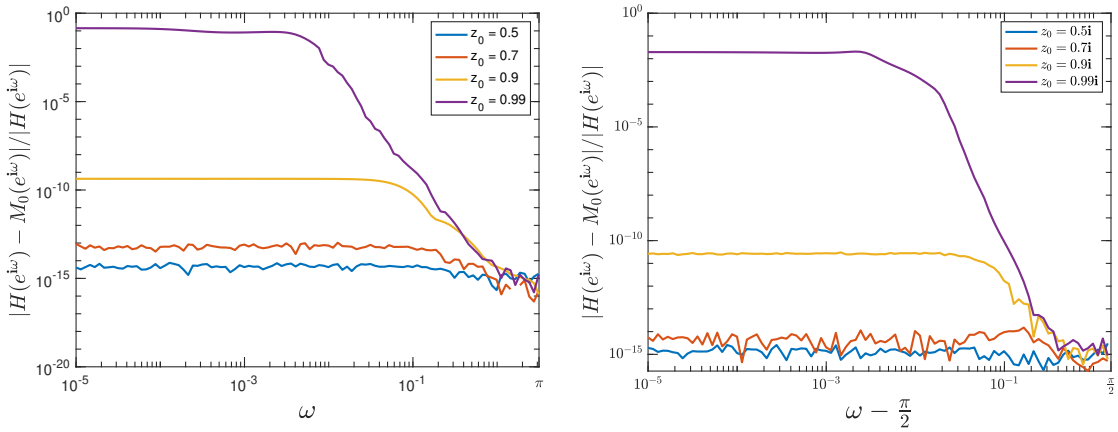
$$H_i(z) = \mathbf{c}^\top (z\mathbf{I} - \mathbf{A}_i)^{-1}\mathbf{b}.$$

The cluster of poles for $\hat{\mathcal{S}}_i$ approaches $\sigma_1 = e^{\mathbf{i}\omega_1} = 1$ (i.e., $\omega_1 = 0$) as i goes from 1 to 4. To illustrate that this cluster can cause errors in learning $M_0(\sigma)$ regardless of the cluster

location, we repeat the same experiment with order $n = 50$ systems $\tilde{\mathcal{S}}_i$, $i = 1, 2, 3, 4$, where every pole μ of $\tilde{\mathcal{S}}_i$ is

$$\mu = \mathbf{i}\lambda,$$

where λ is a pole of $\hat{\mathcal{S}}_i$. So the cluster of poles for $\tilde{\mathcal{S}}_i$ approaches $\sigma_2 = e^{\mathbf{i}\omega_2} = \mathbf{i}$ (i.e., $\omega_2 = \frac{\pi}{2}$) as i goes from 1 to 4.



(a) $\sigma_i = \sigma_1$. (b) $\sigma_i = \sigma_2$.

Figure 4.5: Relative error in $M_0(e^{\mathbf{i}\omega})$ for systems with poles clustered in radius of 0.01 around z_0.

Figure 4.5 shows the effect of moving a pole cluster closer to $\sigma_1 = 1$ (Figure 4.5a) and $\sigma_2 = \mathbf{i}$ (Figure 4.5b). We clearly see that as the cluster of poles approaches σ_i, our approximations of $M_0(e^{\mathbf{i}\omega})$ located near the cluster (i.e., $|\omega - \omega_i| < 10^{-2}$) degrade. This error behavior is not affected by the value of $\hat{n}$ used in calculation of M_0 via (4.44). Even when the true system dimension n is used, we see the same error behavior as in Figure 4.5.

We note that the same behavior is not observed for systems which merely have poles close to interpolation points. That is to say, if we construct a system where the system poles lie *equispaced* on a circle of radius $r < 1$, even when r is very close to 1 (say $r = 0.99$, analogous to Figure 4.5), we are still able to accurately estimate $H(\sigma)$ even for σ close to the poles.

So, it appears that the presence of a *cluster* of poles near σ_i, not just a single pole, is necessary for the observed decrease in accuracy of learned values $M_0(\sigma)$ for σ near σ_i. The reason our method is negatively affected by clustered poles is discussed in Section 4.6.3.

4.6.2 Real Poles

In Section 4.1, we introduced the system $\mathcal{S}_2$ and showed the relationship between the value of $\hat{n}$ used in (4.44) and the relative error in estimating M_0 for $\mathcal{S}_2$ in Figure 4.1b. For $\mathcal{S}_2$, we saw that using $\hat{n}$ found via Theorem 4.1 led to poor mean accuracy in learned M_0 values. We also saw that when we used larger values of $\hat{n}$ (say, the true n), we could attain much higher accuracy in learned M_0 values.

The system $\mathcal{S}_2$ has all real poles. Similar to systems with clustered poles, we observe that systems with all real poles have low accuracy in learned $M_0(\sigma)$ via (4.44) for σ near the real axis (i.e., near the poles of the system) when $\hat{n}$ is found using Theorem 4.1. Unlike systems with clustered poles, we observe that using a value for $\hat{n}$ in (4.44) greater than the value indicated by Theorem 4.1 (e.g., n) yields an increase in accuracy in learned $M_0(\sigma)$.

To illustrate this behavior, we conduct an experiment similar to Section 4.6.1. We construct three systems, $\mathcal{S}_i^r, i = 1, 2, 3$ that all have the same random $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{c}^\top \in \mathbb{R}^{1\times n}$. For the system $\mathcal{S}_i^r$, the matrix $\mathbf{A}_i$ was constructed,

$$\mathbf{A}_i = \mathbf{V}^H \mathbf{D}_i \mathbf{V} \in \mathbb{R}^{n\times n},$$

where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a random orthogonal matrix and $\mathbf{D}_i \in \mathbb{R}^{n\times n}$ is a diagonal matrix with random *real* entries d_j on the diagonal, each satisfying $|d_j| < 1$, $j = 1, \ldots, n$. So for the

system $\mathcal{S}_i^r$ with transfer function $H_i(z)$, we have

$$H_i(z) = \mathbf{c}^\top (z\mathbf{I} - \mathbf{A}_i)^{-1}\mathbf{b}.$$

Figure 4.6 shows the effect of using n instead of $\hat{n}$ obtained via Theorem 4.1 in calculation

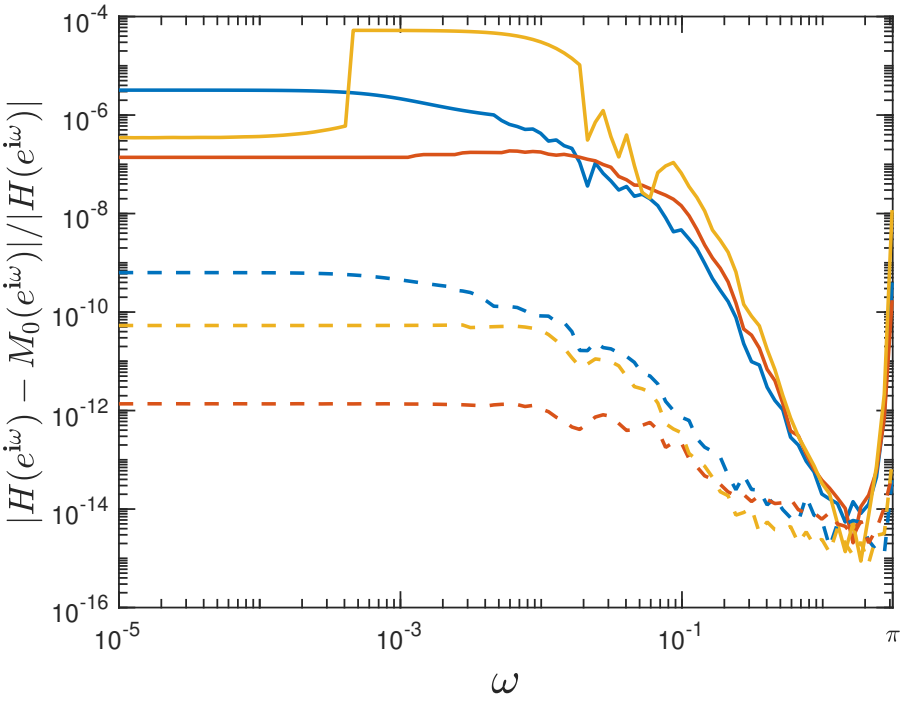


Figure 4.6: Relative errors of learned frequency information for three random systems with all real poles $\mathcal{S}_i^r, i = 1, 2, 3$ for different values of $\hat{n}$ in (4.44). Solid lines represent using $\hat{n}$ calculated using Theorem 4.1 in (4.44) for data generated by $\mathcal{S}_i^r$, the dashed line of the same color represents using $\hat{n} = n$ in (4.44) for data generated by $\mathcal{S}_i^r$. We see that for these systems, using $\hat{n}$ generated by Theorem 4.1 does a poor job of learning M_0 via (4.44), while using n decreases the relative error.

of M_0 using (4.44). We observe a large (four orders of magnitude) decrease in the relative error in approximation of $M_0(\sigma) = M_0(e^{\mathbf{i}\omega})$ for σ near the real axis (or ω near 0 or π). The reason our method is negatively affected by real poles is discussed in Section 4.6.3.

4.6.3 Condition Number

The conditioning of the problem (4.44) is strongly correlated with the relative error in calculating M_0. Indeed, we observe that for the classes of systems described and examined in Sections 4.6.1 and 4.6.2, the condition number of (4.44) increases substantially for σ near a pole cluster and near the real axis, respectively.

To numerically illustrate this, we learn $M_0(\sigma_i) \approx H(\sigma_i)$ at

$$\sigma_i = e^{\mathbf{i}\omega_i},$$

where $\{\omega_i\}_{i=1}^{100}$ are logarithmically spaced in $[10^{-5}, \pi)$. We use $n_w = 20$ windows. We also record the condition number $\kappa_2^k([\ \mathbf{U}_k,\ \mathbf{z}_i])$, where $\mathbf{U}_k$ is the orthogonal subspace for the k-th window ($k = 1, 2, \ldots n_w$) and $\mathbf{z}_i = [\ 0^\top\ \ -\gamma_{\hat{n}}(\sigma_i)^\top]^\top$. As with $M_0(\sigma_i)$, we take

$$\kappa_2([\ \mathbf{U}\ \mathbf{z}_i]) = \frac{1}{n_w}\sum_{k=1}^{n_w} \kappa_2^k([\ \mathbf{U}_k\ \mathbf{z}_i]) \tag{4.45}$$

to be the condition number of calculating $M_0(\sigma_i)$. As usual, we take

$$\epsilon_{rel} = \frac{|M_0(\sigma_i) - H(\sigma_i)|}{|H(\sigma_i)|} \tag{4.46}$$

to be the relative error in calculation of $M_0(\sigma_i)$. We display $10^{-14} \cdot \kappa_2([\ \mathbf{U},\ \mathbf{z}_i])$ versus ϵ_{rel} in Figure 4.7. We observe that the relative error in our learned frequency information closely follows the condition number associated with learning $M_0(\sigma_i)$. This gives us yet another way to estimate the error in our approximations; if the condition number (4.45) is large, we can expect (4.46) to be large as well.

From theory regarding accuracy in computed solutions to linear systems, i.e. that the relative

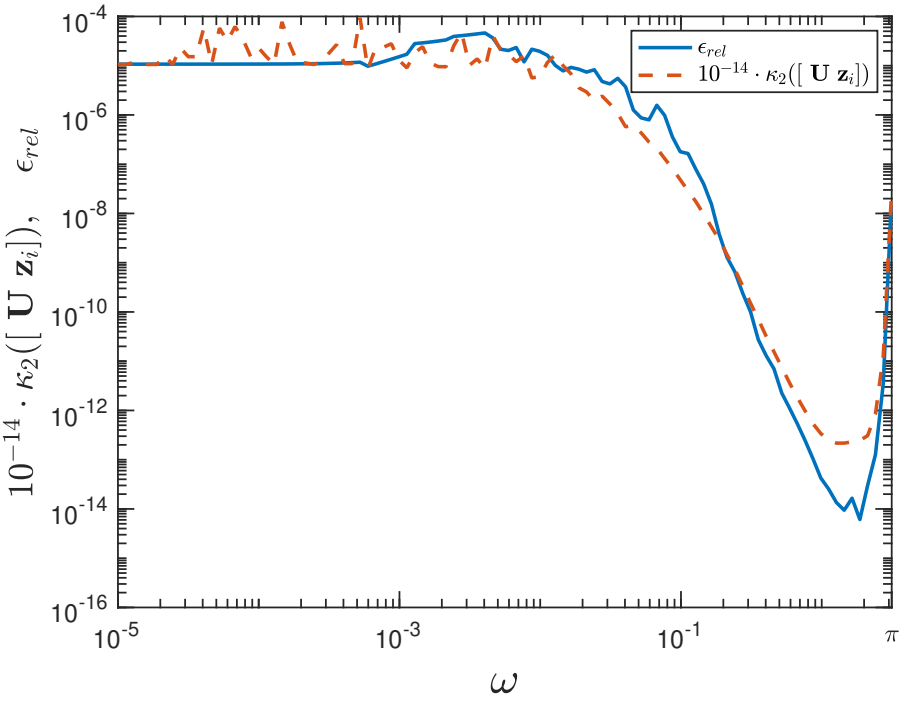


Figure 4.7: The relative error in learned frequency information M_0 closely follows the condition number associated with learning M_0.

error in the computed solution to a linear system is directly proportional to the condition number [12, 27], we can say with confidence that this condition number is the main driver of the error seen in estimating $H(\sigma)$. The remaining questions are (1) what about a system will cause the problem (4.44) to become ill-conditioned for certain σ, and (2) why do these features lead to ill-conditioning?

While Sections 4.6.1 and 4.6.2 give partial answers to question (1), they do not provide insight into question (2). These sections also raise another question, (3) why does using $\hat{n}$ obtained from Theorem 4.1 lead to low accuracy in M_0 for certain $\sigma \in \mathbb{C}$ learned via (4.44) for both systems with clustered poles and real poles, but using n only provides an increase in accuracy for systems with real poles? Questions (2) and (3) are as of now still unanswered, and are an area for future research. Answering these questions could indicate the next steps

to making the data informativity framework more robust and applicable to a wider class of systems.

Chapter 5

Implementing Frequency Methods from Time Domain Data

The results of Chapter 4 allow us to calculate values and derivatives of transfer functions for a large number of systems. We now test how well this estimated frequency domain data can be used to create ROMs from established frequency domain techniques. In particular, we develop Time Domain (TD)-IRKA, an extension of TF-IRKA (Algorithm 1) to the case where we have access to only one time domain simulation of a system.

5.1 Time Domain Iterative Rational Krylov Algorithm

Recall Algorithm 1, one step of which was to calculate $H(\sigma)$ and $H'(\sigma)$. We can now use the results of Chapter 4 to calculate this information from time domain data. We refer to this extension as the Time Domain (TD)-IRKA algorithm (Algorithm 3). We emphasize that the only difference between Algorithms 3 and 1 is how we obtain new frequency domain information at each step. Algorithm 1 assumes the ability to calculate the required information directly while Algorithm 3 learns the information from time domain data.

We emphasize that the advantage of using the process described in Chapters 3 and 4 in Algorithm 3 is the quick, low cost estimation of frequency information $H(\sigma)$ from time domain data $\mathbb{U}$ and $\mathbb{Y}$. This allows for construction of $\mathcal{H}_2$ optimal ROMs from only a single

Algorithm 3 TD-IRKA

Require: $\{\sigma_i\}_{i=1}^r$, an initial set of interpolation points closed under conjugation (i.e., if σ is an interpolation point, so is $\overline{\sigma}$).
Learn H and H' at $\{\sigma_i\}_{i=1}^r$ via the process described in Chapters 3 and 4.
Form $\mathbf{E}_\rho$ and $\mathbf{A}_\rho$ using σ_i, $H(\sigma_i)$, and $H'(\sigma_i)$ from (2.9).
while Not Converged **do**
  Calculate generalized eigenvalues $\{\lambda_i\}_{i=1}^r$ of the pencil $\lambda\mathbf{E}_\rho - \mathbf{A}_\rho$
  $\sigma_i \leftarrow \frac{1}{\lambda_i}$
  Relearn H and H' at new $\{\sigma_i\}_{i=1}^r$ via the process described in Chapters 3 and 4.
  Form new $\mathbf{E}_\rho$ and $\mathbf{A}_\rho$ from (2.9) using σ_i, $H(\sigma_i)$, and $H'(\sigma_i)$.
end while
Form $\mathbf{E}_\rho, \mathbf{A}_\rho, \mathbf{B}_\rho, \mathbf{C}_\rho$ from (2.9).

time domain input trajectory and associated output trajectory.

5.2 Convergence Rate Compared to TF-IRKA

In this section, we examine how the relative $\mathcal{H}_2$ error (Section 1.4) in TD-IRKA ROMs decay as the order of the ROM r increases by comparing to TF-IRKA. We compare performance on the two systems, $\mathcal{S}_1$ and $\mathcal{S}_2$, introduced in Section 4.1. We seek ROMs of order

$$r = 2, 4, 6, \ldots, 30,$$

and initialize both algorithms with equispaced poles

$$\eta_k = 1.5 \cdot e^{\frac{2k\pi\mathbf{i}}{r}}, \quad k = 1, 2, \ldots, r. \tag{5.1}$$

TF-IRKA has access to functions that evaluate the transfer function to machine precision, while TD-IRKA only has access to a single time domain simulation of the system. The relative $\mathcal{H}_2$ errors of ROMs formed from TD-IRKA and TF-IRKA to approximate $\mathcal{S}_1$ are shown in Figure 5.1. We observe in Figure 5.1 that TD-IRKA follows the convergence of

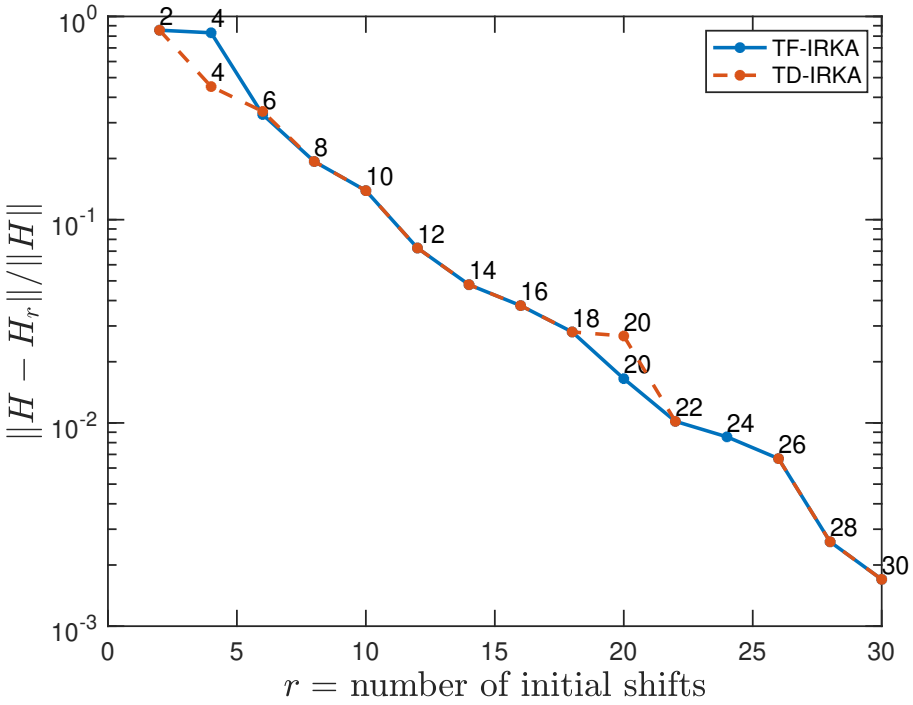


Figure 5.1: Relative $\mathcal{H}_2$ errors of ROMs of increasing order approximating $\mathcal{S}_1$. The dimension of the returned ROM is shown above each data point.

TF-IRKA very closely. We also observe that for each r, both methods returned functions that were order r. This is not always guaranteed, as the Loewner systems constructed at each step could have truncated the order to $\rho < r$. To confirm that both TD-IRKA and TF-IRKA converge to the same reduced order model, we plot the converged TF-IRKA and TD-IRKA shifts for the order $r = 14$ ROM in Figure 5.2. We observe that both algorithms were converging to the same $\mathcal{H}_2$ optimal ROM.

Next, we examine the convergence of both algorithms on $\mathcal{S}_2$. Figure 5.3 shows a very different story than Figure 5.1. We observe that TD-IRKA only tracks the convergence of TF-IRKA until $r = 8$. At $r = 10$, TF-IRKA produces an order 8 approximation, which actually performs worse than TD-IRKA. Then from $r = 12$ on, neither method makes any gains in accuracy, with the minimum $\mathcal{H}_2$ error being 6.88×10^{-6} and 1.93×10^{-6} for TD-IRKA and

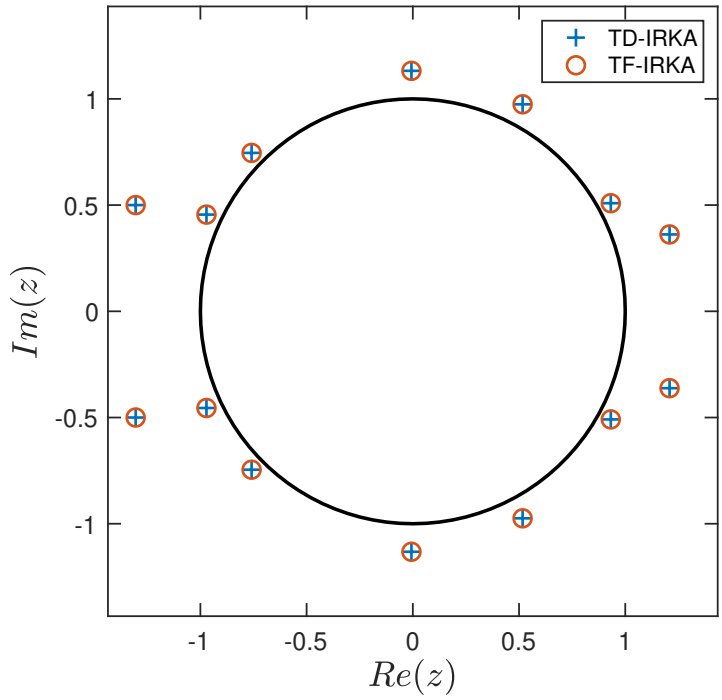


Figure 5.2: Converged TD-IRKA and TF-IRKA shifts for an order $r = 14$ ROM.

TF-IRKA, respectively. This leveling off is due to the ROMs not exceeding order 12.

In Section 5.3, we will again form ROMs using $\mathcal{S}_2$, using the same set up as in the current section. However, the relative error that TD-IRKA levels off at in Figure 5.3 (6.88×10^{-6}) does *not* match the relative error (1.8312×10^{-4}) for a TD-IRKA ROM of $\mathcal{S}_2$ with $r = 16$ initial shifts displayed in Table 5.4 of Section 5.3. No changes were made to the implementation between generating this data. The only difference is that the input $\mathbb{U}$ was a different Gaussian random noise signal. The small changes in approximations to M_0 and M_1 caused by different input-output data led the first Hermite Loewner approximation in TD-IRKA to return an order $\rho = 10$ system for the data used to generate Table 5.4, while an order $\rho = 12$ system was returned for the data used to generate Figure 5.3.

In Figure 5.4, we plot the convergence of TD-IRKA and TF-IRKA for a different input

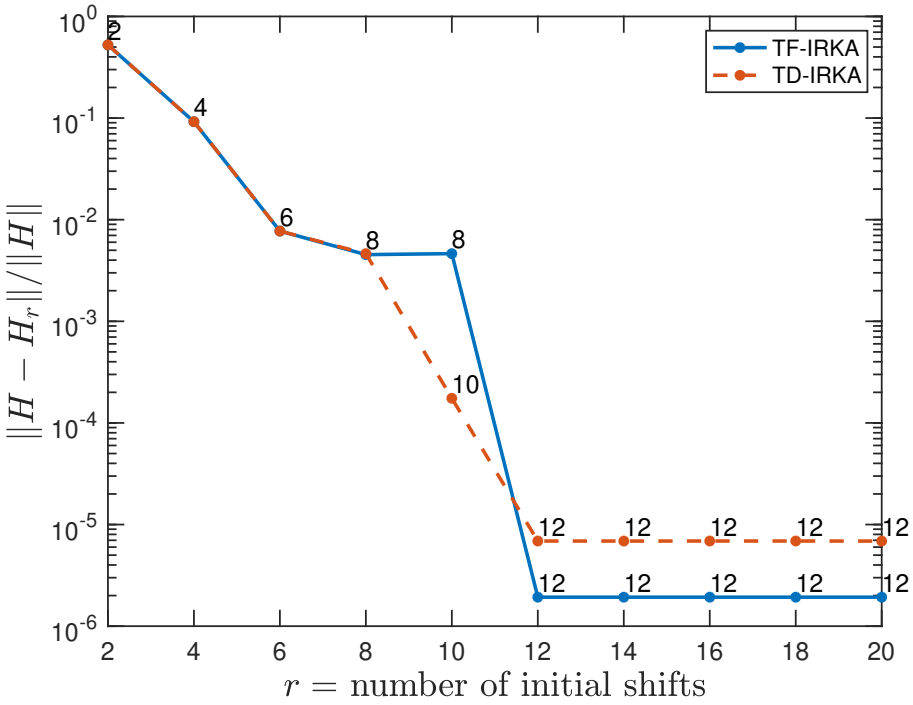


Figure 5.3: Relative $\mathcal{H}_2$ errors of ROMs of increasing order approximating $\mathcal{S}_2$. The dimension of the returned ROM is shown above each data point. Order $\rho < r$ ROMs are possible due to the use of the Loewner Framework.

$\hat{\mathbb{U}}$ than in Figure 5.3. This time, we see that the ROMs produced by TD-IRKA level off starting at $r = 10$, with a similar $\mathcal{H}_2$ error to Table 5.4. To check that TD-IRKA converged to a locally optimal order 10 model (and thus confirm that the loss of degrees of freedom was the reason for larger error), we initialize TF-IRKA with the reciprocals of the poles of the converged TD-IRKA model. More precisely, if $\tilde{H}_l^{TD}$ is the ROM produced by TD-IRKA, and λ_k is the k-th pole of $\tilde{H}_l^{TD}$, then we initialize TF-IRKA with

$$\eta_k = \frac{1}{\lambda_k}, \quad k = 1, 2, \ldots, 10,$$

to produce $\tilde{H}^{TF}$, an order 10 TF-IRKA ROM using true data. The results of this test, shown in Table 5.1, indicate that TD-IRKA did indeed converge to a ROM that is near to a locally

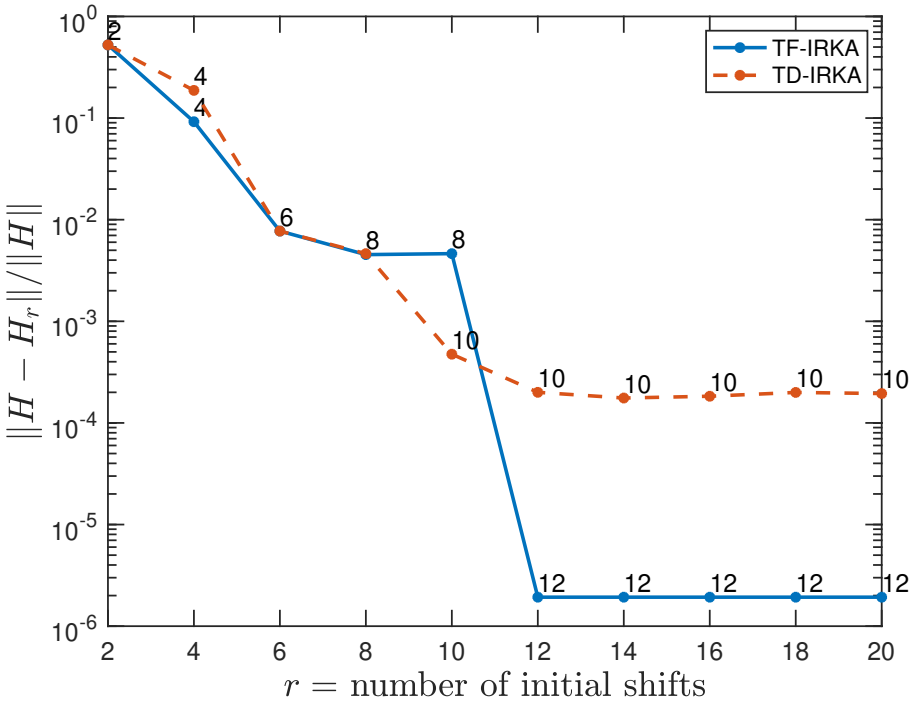


Figure 5.4: Relative $\mathcal{H}_2$ errors of ROMs of increasing order approximating $\mathcal{S}_2$, with a different input than used in Figure 5.3. The dimension of the returned ROM is shown above each data point. Order $\rho < r$ ROMs are possible due to the use of the Loewner Framework.

$\mathcal{H}_2$ optimal order 10 ROM. Thus, the main driver in the loss of accuracy was the truncation step in Hermite Loewner. These results motivate finding ways to predict when Hermite

Table 5.1: Relative $\mathcal{H}_2$ errors of the TD-IRKA ROM produced from starting poles specified in (5.5) and the nearest true TF-IRKA ROM.

$\frac{\|H - \tilde{H}_l^{TD}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}}$	$\frac{\|H - \tilde{H}^{TF}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}}$	$\frac{\|H^{TF} - \tilde{H}_l^{TD}\|_{\mathcal{H}_2}}{\|H^{TF}\|_{\mathcal{H}_2}}$
1.945×10^{-4}	1.747×10^{-4}	8.0508×10^{-5}

Loewner will return ROMs with smaller order than expected. In [15], the authors suggest that the decay of the singular values of Loewner matrices is very sensitive to the choice of interpolation points. Further, it is shown in [4] that TF-IRKA (and thus also TD-IRKA) has a remarkable ability to choose interpolation points that produce slow decay in the singular

values of Loewner matrices. Thus, a major next step for this project is to find a good (well conditioned) set of initial poles, so that the singular values of the Loewner matrices decay slowly even in the first steps of TF-IRKA and TD-IRKA.

5.3 Comparison to Other ROM Methods

In this section, we compare TD-IRKA's performance at creating a ROM to Loewner (Section 2.1), VF (Section 2.2.1), and AAA (Section 2.2.2) for two test systems, $\mathcal{S}_1$ and $\mathcal{S}_2$. These systems were introduced in Section 4.1. Both systems are real, with 100 random stable poles, closed under conjugation.

The transfer function $H(z)$ and (when needed) $H'(z)$ were sampled at 100 logrithmicaly spaced frequencies

$$\{\omega_i\}_{i=1}^{100}, \qquad \omega_i \in [10^{-5}, \pi).$$

These frequencies correspond to 100 points in the top half unit circle,

$$\{\tilde{\sigma}_i\}_{i=1}^{100}, \qquad \tilde{\sigma}_i = e^{\mathbf{i}\omega_i}.$$

Then, since both systems are real, we immediately obtain conjugate data at $\{\overline{\tilde{\sigma}}_i\}_{i=1}^{100}$, and our complete set of points where we have knowledge of the transfer function is

$$\{\sigma_i\}_{i=1}^{200} = \{\tilde{\sigma}_i\}_{i=1}^{100} \cup \{\overline{\tilde{\sigma}}_i\}_{i=1}^{100}. \tag{5.2}$$

So, we have access to

$$\begin{aligned} H(\sigma_i) &\approx M_0(\sigma_i), \quad i = 1, 2, \ldots, 200 \\ H'(\sigma_i) &\approx M_1(\sigma_i), \quad i = 1, 2, \ldots, 200. \end{aligned} \tag{5.3}$$

ROMs from three different methods, Loewner Framework, Vector Fitting, and AAA were constructed using both true frequency domain data $(\sigma_i, H(\sigma_i), H'(\sigma_i))$ obtained directly from the system matrices, and learned frequency domain data $(\sigma_i, M_0(\sigma_i), M_1(\sigma_i))$ obtained through Algorithm 2. These six resulting ROMs were then compared to a ROM produced by TF-IRKA using true frequency domain data, and a ROM produced by TD-IRKA, which had access to only one time domain simulation of the system.

Vector Fitting was given initial poles

$$p_k = e^{\frac{2k\pi \mathbf{i}}{r}}, \quad k = 1, 2, \ldots, r, \tag{5.4}$$

while TF-IRKA and TD-IRKA were given initial poles

$$\eta_k = 1.5 \cdot p_k. \tag{5.5}$$

For both systems we seek an order $r = 16$ ROM using each of the above methods.

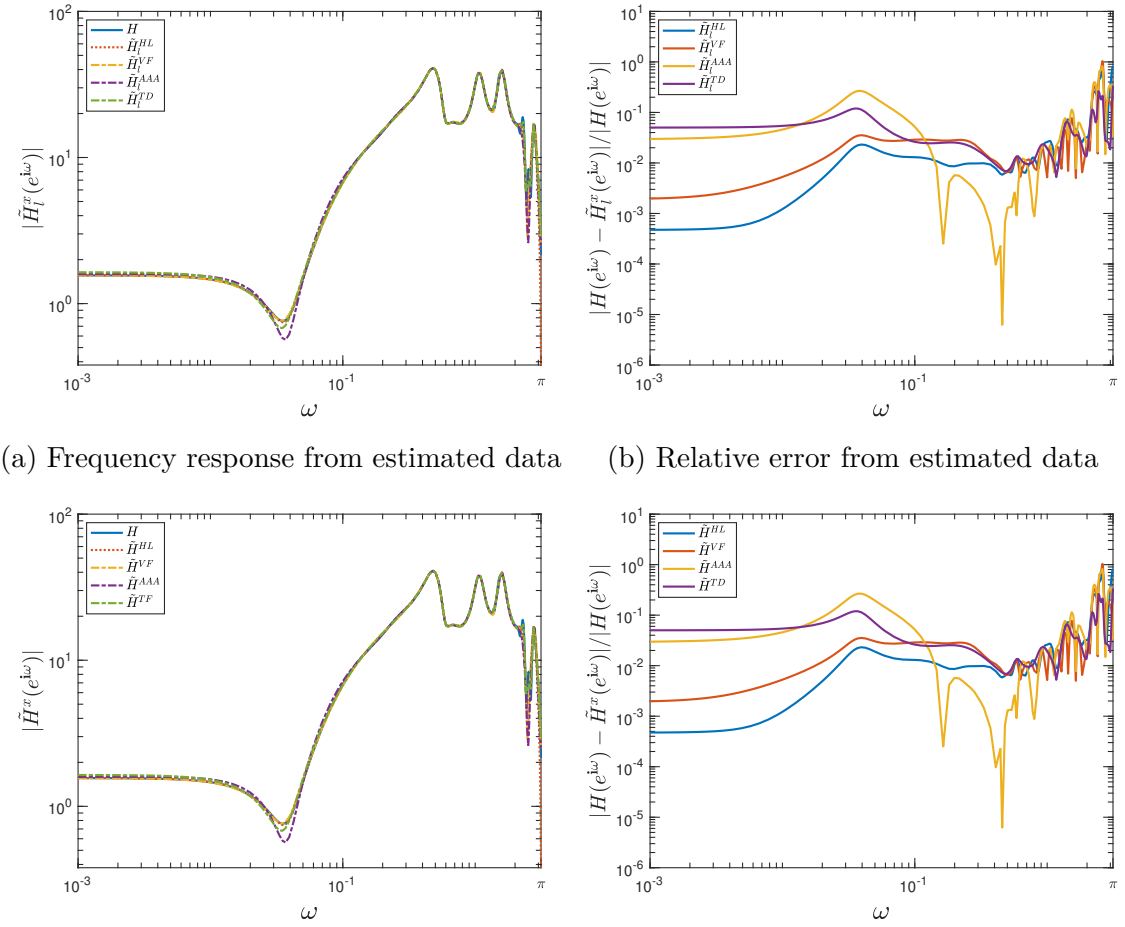
5.3.1 A well-behaved example

For the first system, $\mathcal{S}_1$ with transfer function $H(z)$, we calculate $\hat{n} = 85$ using Theorem 4.1. This $\hat{n}$ allows us to accurately learn M_0 and M_1, with mean relative errors

$$\frac{|M_0(\sigma_i) - H(\sigma_i)|}{|H(\sigma_i)|} \approx \frac{|M_1(\sigma_i) - H'(\sigma_i)|}{|H'(\sigma_i)|} \approx 10^{-10}.$$

This was shown in Section 4.1. So, we expect ROMs produced from M_0 and M_1 to not be far off from ROMs produced from $H(\sigma)$ and $H'(\sigma)$. We refer to a ROM from true data by $\tilde{H}$ and a ROM from learned data by $\tilde{H}_l$.

The frequency responses of the Loewner, Vector Fitting, AAA, and IRKA models, as well as their relative errors are displayed in Figure 5.5.



(a) Frequency response from estimated data

(b) Relative error from estimated data

(c) Frequency response from true data

(d) Relative error from true data

Figure 5.5: Frequency response and relative errors of ROMs from true and measured data

At first glance, we see that none of the ROMs were qualitatively affected by errors in the calculation of $M_0(\sigma_i)$ and $M_1(\sigma_i)$. The resulting $\mathcal{H}_2$ errors, $\mathcal{H}_\infty$ errors, and the relative difference between the ROMs is displayed in Table 5.2 and Table 5.3, respectively.

We observe that none of the ROM methods were drastically effected by the errors introduced by learning frequency domain data from time domain data. We also observe that both TD-IRKA and TF-IRKA beat the other ROMs by an order of magnitude in both a relative $\mathcal{H}_2$

Table 5.2: $\mathcal{H}_2$ errors between the true transfer function H, ROMs constructed from true data $\tilde{H}$, and ROMs constructed from learned data $\tilde{H}_l$

	Loewner	VF	AAA	IRKA
$\frac{\|H-\tilde{H}_l\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}}$	8.2160×10^{-2}	9.2729×10^{-2}	1.0091×10^{-1}	3.7786×10^{-2}
$\frac{\|H-\tilde{H}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}}$	8.2160×10^{-2}	9.2729×10^{-2}	1.0091×10^{-1}	3.7786×10^{-2}
$\frac{\|\tilde{H}-\tilde{H}_l\|_{\mathcal{H}_2}}{\|\tilde{H}\|_{\mathcal{H}_2}}$	1.0735×10^{-10}	2.1847×10^{-11}	6.1155×10^{-11}	2.8114×10^{-8}

Table 5.3: $\mathcal{H}_\infty$ errors between the true transfer function H, ROMs constructed from true data $\tilde{H}$, and ROMs constructed from learned data $\tilde{H}_l$

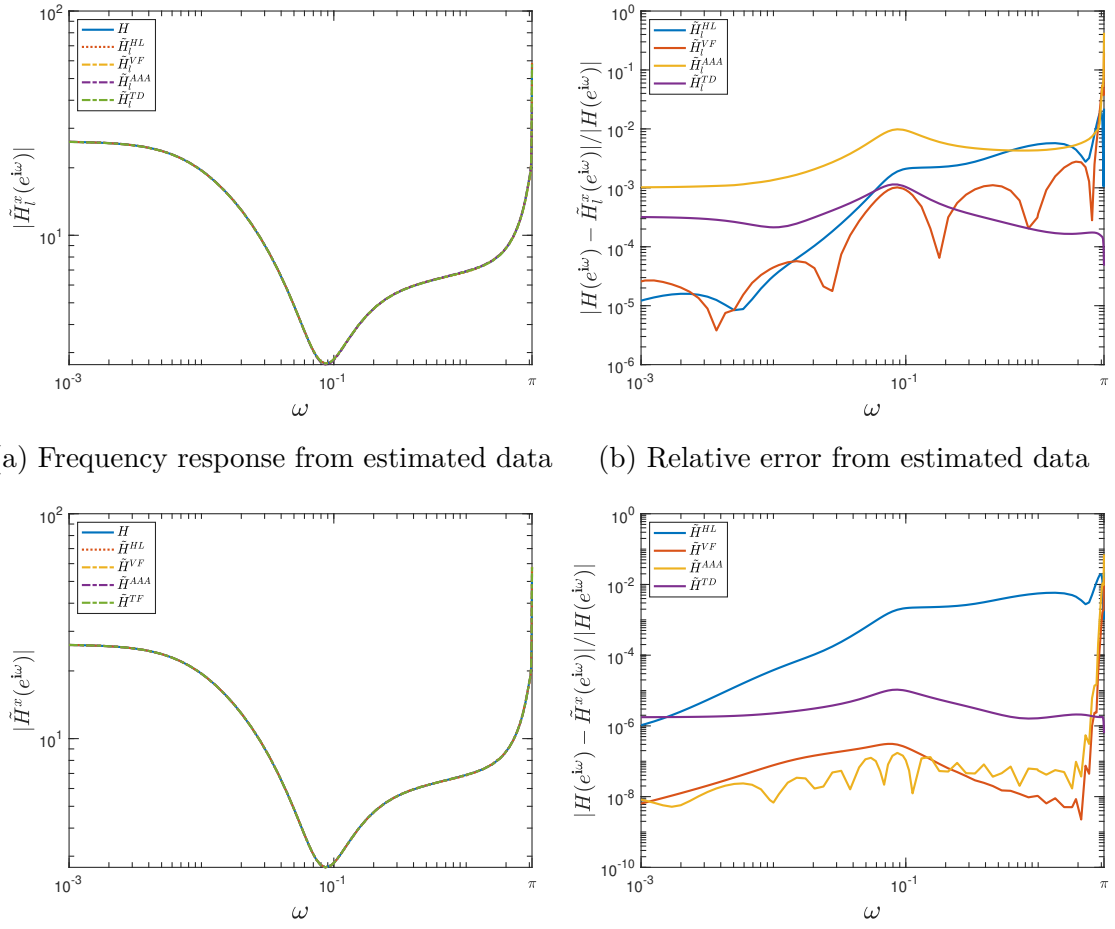
	Loewner	VF	AAA	IRKA
$\frac{\|H-\tilde{H}_l\|_{\mathcal{H}_\infty}}{\|H\|_{\mathcal{H}_\infty}}$	1.1933×10^{-1}	1.5079×10^{-1}	1.8326×10^{-1}	4.2931×10^{-2}
$\frac{\|H-\tilde{H}\|_{\mathcal{H}_\infty}}{\|H\|_{\mathcal{H}_\infty}}$	1.1933×10^{-1}	1.5079×10^{-1}	1.8326×10^{-1}	4.2931×10^{-2}
$\frac{\|\tilde{H}-\tilde{H}_l\|_{\mathcal{H}_\infty}}{\|\tilde{H}\|_{\mathcal{H}_\infty}}$	1.7853×10^{-10}	3.6597×10^{-11}	7.8475×10^{-11}	4.800×10^{-8}

sense and relative $\mathcal{H}_\infty$ sense. In Section 5.3.2, we investigate how ROMs perform on less accurate learned frequency data.

5.3.2 An ill-behaved example

For the second system, $\mathcal{S}_2$ with transfer function $H(z)$, we calculate $\hat{n} = 25$. Recall from Figure 4.1b that this $\hat{n}$ still produces large $\mathcal{O}(10^{-5})$ relative errors in learned frequency domain data. Thus, we anticipate ROMs approximating $\mathcal{S}_2$ could perform significantly worse when constructed with $M_0(\sigma_i)$ and $M_1(\sigma_i)$ instead of $H(\sigma_i)$ and $H'(\sigma_i)$.

The frequency responses of the Loewner, Vector Fitting, AAA, and IRKA ROMs, as well as their relative errors are displayed in Figure 5.6.



(a) Frequency response from estimated data

(b) Relative error from estimated data

(c) Frequency response from true data

(d) Relative error from true data

Figure 5.6: Frequency response and relative errors of ROMs from true and measured data for $\mathcal{S}_2$

In contrast to $\mathcal{S}_1$, we see that some ROMs approximating $\mathcal{S}_2$ are significantly affected by the errors in the learned data. The $\mathcal{H}_2$ errors, $\mathcal{H}_\infty$ errors, and ROM relative differences (Table 5.4 and 5.5) show significant differences in the VF and AAA ROMs. The IRKA ROMs were also affected, although to a lesser degree than VF and AAA. We also remark that the AAA ROM made from learned data failed to produce a stable system when using learned data, so

the unstable poles were removed from the system. We observe that once again, both IRKA models significantly outperformed the other three ROM techniques in both the relative $\mathcal{H}_2$ and relative $\mathcal{H}_\infty$ sense.

Table 5.4: $\mathcal{H}_2$ errors between the true transfer function H, ROMs constructed from true data $\tilde{H}$, and ROMs constructed from learned data $\tilde{H}_l$

	Loewner	VF	AAA	IRKA
$\frac{\|H-\tilde{H}_l\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}}$	1.0773×10^{-2}	5.7539×10^{-2}	1.3038×10^{-1}	1.8312×10^{-4}
$\frac{\|H-\tilde{H}\|_{\mathcal{H}_2}}{\|H\|_{\mathcal{H}_2}}$	1.0739×10^{-2}	4.0697×10^{-3}	1.8964×10^{-2}	1.929×10^{-6}
$\frac{\|\tilde{H}-\tilde{H}_l\|_{\mathcal{H}_2}}{\|\tilde{H}\|_{\mathcal{H}_2}}$	8.711×10^{-5}	6.1085×10^{-2}	1.1613×10^{-1}	1.8311×10^{-4}

Table 5.5: $\mathcal{H}_\infty$ errors between the true transfer function H, ROMs constructed from true data $\tilde{H}$, and ROMs constructed from learned data $\tilde{H}_l$

	Loewner	VF	AAA	IRKA
$\frac{\|H-\tilde{H}_l\|_{\mathcal{H}_\infty}}{\|H\|_{\mathcal{H}_\infty}}$	2.2069×10^{-2}	1.9527×10^{-1}	4.3828×10^{-1}	1.4589×10^{-4}
$\frac{\|H-\tilde{H}\|_{\mathcal{H}_\infty}}{\|H\|_{\mathcal{H}_\infty}}$	2.1976×10^{-2}	1.5711×10^{-2}	7.1022×10^{-2}	7.9654×10^{-7}
$\frac{\|\tilde{H}-\tilde{H}_l\|_{\mathcal{H}_\infty}}{\|\tilde{H}\|_{\mathcal{H}_\infty}}$	9.0753×10^{-5}	2.0772×10^{-1}	4.0388×10^{-1}	1.451×10^{-4}

While TD-IRKA performed quite well on both $\mathcal{S}_1$ and $\mathcal{S}_2$, there is still room for improvement. The main area for improvement is reducing Loewner truncation. Since TD-IRKA uses the Loewner framework at every step, which can result in ROMs with order $\rho < r$ (see Section 2.1), TD-IRKA can produce an order $\rho < r$ approximation. Generally, an IRKA ROM with fewer degrees of freedom will have greater $\mathcal{H}_2$ error. Indeed, when seeking an order $r = 16$ approximation for $\mathcal{S}_2$, TD-IRKA produced an order $\rho = 10$ approximation

and TF-IRKA produced an order $\rho = 12$ approximation. If we were able to find an order $r = 16$ ROM with these methods they would perform even better than they already do when compared to Loewner, VF, and AAA. As discussed in Section 5.2 and shown in [15], some sets of starting shifts may be less likely to cause truncation in the Loewner interpolation step, so choosing a better set of starting shifts could achieve our goal of constructing ROMs without truncation to $\rho < r$.

Chapter 6

Conclusions and Future Work

Frequency based ROM techniques are valuable tools for computing ROMs of complex systems. However, sometimes frequency information is unavailable. Burohman et al. [11] introduced the data informativity framework to learn frequency information from purely time domain data. In this thesis, we provided several key updates to the framework, geared towards a fast and robust numerical implementation. We then used this numerical implementation to develop TD-IRKA, extending the ability to construct $\mathcal{H}_2$ optimal reduced order models to the case where only time domain data is available. We compared TD-IRKA to TF-IRKA, which showed that TD-IRKA can display similar performance behavior to TF-IRKA, especially for small reduced order r. Finally, we investigated how well established frequency domain techniques (Loewner, Vector Fitting, and AAA) performed on our learned frequency domain data while comparing performance to TD-IRKA. This comparison showed that TD-IRKA can outperform other frequency based ROM methods using learned data in a relative $\mathcal{H}_2$ sense.

There are many directions for future work on this project. One major direction is investigating the utility of multiple inputs (i.e., using more than one set of input-output data $\mathbb{U}, \mathbb{Y}$). Another direction is to learn why clustered poles and poles on the real axis negatively effect our ability to learn $M_0(\sigma)$ for σ close to a pole cluster or the real axis, respectively. We also hope to apply and extend the results of [15] to find a more stable set of starting shifts for TD-IRKA, in order to reduce the likelihood of constructing an order $\rho < r$ ROM

when r initial shifts are used. Finally, we hope to be able to extend the data informativity framework to continuous time systems and to nonlinear systems.

Bibliography

[1] Suliman Al-Homidan, Mohammad M. Alshahrani, Cosmin G. Petra, and Florian A. Potra. Minimal condition number for positive definite Hankel matrices using semidefinite programming. *Linear Algebra and its Applications*, 433(6):1101–1109, November 2010. ISSN 00243795. doi: 10.1016/j.laa.2010.04.052. URL https://linkinghub.elsevier.com/retrieve/pii/S0024379510002958.

[2] A.C. Antoulas, C.A. Beattie, and Gügercin. *Interpolary Methods for Model Reduction*. Computational Science and Engineering. SIAM, 2020. ISBN 978-1-61197-607-6.

[3] Athanasios Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005. ISBN 978-0-89871-658-0.

[4] C. Beattie, Z. Drmač, and S. Gugercin. Revisiting IRKA: Connections with pole placement and backward stability. *arXiv:1911.05804 [cs, eess, math]*, November 2019. URL http://arxiv.org/abs/1911.05804. arXiv: 1911.05804.

[5] Christopher Beattie and Serkan Gugercin. Realization-independent H2-approximation. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 4953–4958, Maui, HI, USA, December 2012. IEEE. ISBN 978-1-4673-2066-5 978-1-4673-2065-8 978-1-4673-2063-4 978-1-4673-2064-1. doi: 10.1109/CDC.2012.6426344. URL http://ieeexplore.ieee.org/document/6426344/.

[6] Bernhard Beckermann. The condition number of real Vandermonde, Krylov and positive definite Hankel matrices. *Numerische Mathematik*, 85(4):553–577, June 2000. ISSN 0029-599X. doi: 10.1007/PL00005392. URL http://link.springer.com/10.1007/PL00005392.

[7] Peter Benner, Serkan Gugercin, and Karen Willcox. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review*, 57(4): 483–531, January 2015. ISSN 0036-1445, 1095-7200. doi: 10.1137/130932715. URL http://epubs.siam.org/doi/10.1137/130932715.

[8] Peter Benner, Albert Cohen, Mario Ohlberger, and Karen Willcox. *Model Reduction and Approximation: Theory and Algorithms*. Computational Science and Engineering. SIAM, 2017. ISBN 978-1-61197-481-2.

[9] Mario Berljafa and Stefan Güttel. The RKFIT Algorithm for Nonlinear Rational Approximation. *SIAM Journal on Scientific Computing*, 39(5):A2049–A2071, January 2017. ISSN 1064-8275, 1095-7197. doi: 10.1137/15M1025426. URL https://epubs.siam.org/doi/10.1137/15M1025426.

[10] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[11] Azka Muji Burohman, Bart Besselink, Jacquelien M. A. Scherpen, and M. Kanat Camlibel. From data to reduced-order models via moment matching. *arXiv:2011.00150 [cs, eess, math]*, October 2020. URL http://arxiv.org/abs/2011.00150. arXiv: 2011.00150.

[12] James Demmel. *Applied Numerical Linear Algebra*. Other Titles in Applied Mathematics. SIAM, 1997. ISBN 978-0-89871-389-3.

[13] Z. Drmač, S. Gugercin, and C. Beattie. Quadrature-Based Vector Fitting for Discretized H2 Approximation. *SIAM Journal on Scientific Computing*, 37(2):A625–A652, January 2015. ISSN 1064-8275, 1095-7197. doi: 10.1137/140961511. URL http://epubs.siam.org/doi/10.1137/140961511.

[14] Z Drmač, S Gugercin, and C Beattie. Vector Fitting for Matrix-Valued Rational Approximation. *SIAM Journal on Scientific Computing*, 37(5):34, 2015. doi: 10.1137/15M1010774.

[15] Mark Embree and A. Cosmin Ionita. Pseudospectra of Loewner Matrix Pencils. *arXiv:1910.12153 [cs, math]*, October 2019. URL http://arxiv.org/abs/1910.12153. arXiv: 1910.12153.

[16] Silviu-Ioan Filip, Yuji Nakatsukasa, Lloyd N. Trefethen, and Bernhard Beckermann. Rational Minimax Approximation via Adaptive Barycentric Representations. *SIAM Journal on Scientific Computing*, 40(4):A2427–A2455, January 2018. ISSN 1064-8275, 1095-7197. doi: 10.1137/17M1132409. URL https://epubs.siam.org/doi/10.1137/17M1132409.

[17] Ion Victor Gosea and Serkan Gugercin. The AAA framework for modeling linear dynamical systems with quadratic output. *arXiv:2005.10316 [cs, eess, math]*, May 2020. URL http://arxiv.org/abs/2005.10316. arXiv: 2005.10316.

[18] 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(2): 609–638, January 2008. ISSN 0895-4798, 1095-7162. doi: 10.1137/060666123. URL http://epubs.siam.org/doi/10.1137/060666123.

[19] B. Gustavsen and A. Semlyen. Simulation of transmission line transients using vector fitting and modal decomposition. *IEEE Transactions on Power Delivery*, 13(2):605–614, April 1998. ISSN 08858977. doi: 10.1109/61.660941. URL http://ieeexplore.ieee.org/document/660941/.

[20] Jeffrey M. Hokanson. Projected Nonlinear Least Squares for Exponential Fitting. *SIAM Journal on Scientific Computing*, 39(6):A3107–A3128, January 2017. ISSN 1064-8275,

1095-7197. doi: 10.1137/16M1084067. URL https://epubs.siam.org/doi/10.1137/16M1084067.

[21] Jeffrey M. Hokanson and Caleb C. Magruder. Least Squares Rational Approximation. *arXiv:1811.12590 [math]*, November 2018. URL http://arxiv.org/abs/1811.12590.

[22] Jeffrey M. Hokanson and Caleb C. Magruder. H2-Optimal Model Reduction Using Projected Nonlinear Least Squares. *SIAM Journal on Scientific Computing*, 42(6): A4017–A4045, January 2020. ISSN 1064-8275, 1095-7197. doi: 10.1137/19M1247863. URL https://epubs.siam.org/doi/10.1137/19M1247863.

[23] A. Cosmin Ionita and Athanasios C. Antoulas. Matrix pencils in time and frequency domain system identification. In Li Qiu, Jie Chen, Tetsuya Iwasaki, and Hisaya Fujioka, editors, *Developments in Control Theory Towards Glocal Control*, pages 79–88. Institution of Engineering and Technology, January 2012. ISBN 978-1-84919-533-1 978-1-84919-534-8. doi: 10.1049/PBCE076E_ch9. URL https://digital-library.theiet.org/content/books/10.1049/pbce076e_ch9.

[24] J. Nathan Kutz, Steven L. Brunton, Bingni W. Brunton, and Joshua L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, November 2016. ISBN 978-1-61197-449-2 978-1-61197-450-8. doi: 10.1137/1.9781611974508. URL http://epubs.siam.org/doi/book/10.1137/1.9781611974508.

[25] Sanda Lefteriu and Athanasios C. Antoulas. On the Convergence of the Vector-Fitting Algorithm. *IEEE Transactions on Microwave Theory and Techniques*, 61(4):1435–1443, April 2013. ISSN 0018-9480, 1557-9670. doi: 10.1109/TMTT.2013.2246526. URL http://ieeexplore.ieee.org/document/6470726/.

[26] Pieter Lietaert, Javier Pérez, Bart Vandereycken, and Karl Meerbergen. Automatic rational approximation and linearization of nonlinear eigenvalue problems. *arXiv:1801.08622 [math]*, February 2018. URL http://arxiv.org/abs/1801.08622. arXiv: 1801.08622.

[27] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, 1997. ISBN 978-0-89871-361-9.

[28] A.J. Mayo and A.C. Antoulas. A framework for the solution of the generalized realization problem. *Linear Algebra and its Applications*, 425(2-3):634–662, September 2007. ISSN 00243795. doi: 10.1016/j.laa.2007.03.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0024379507001280.

[29] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, February 1981. ISSN 0018-9286. doi: 10.1109/TAC.1981.1102568. URL http://ieeexplore.ieee.org/document/1102568/.

[30] Yuji Nakatsukasa, Olivier Sète, and Lloyd N. Trefethen. The AAA Algorithm for Rational Approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, January 2018. ISSN 1064-8275, 1095-7197. doi: 10.1137/16M1106122. URL https://epubs.siam.org/doi/10.1137/16M1106122.

[31] Benjamin Peherstorfer, Serkan Gugercin, and Karen Willcox. Data-Driven Reduced Model Construction with Time-Domain Loewner Models. *SIAM Journal on Scientific Computing*, 39(5):A2152–A2178, January 2017. ISSN 1064-8275, 1095-7197. doi: 10.1137/16M1094750. URL https://epubs.siam.org/doi/10.1137/16M1094750.

[32] Alexander D Poularikas. The Z-Transform. In *The Transforms and Applications Handbook: Second Edition*. CRC Press LLC, 2020.

[33] Elizabeth Qian, Boris Kramer, Benjamin Peherstorfer, and Karen Willcox. Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, May 2020. ISSN 01672789. doi: 10.1016/j.physd.2020.132401. URL https://linkinghub.elsevier.com/retrieve/pii/S0167278919307651.

[34] P. Rajendra and V. Brahmajirao. Modeling of dynamical systems through deep learning. *Biophysical Reviews*, 12(6):1311–1320, December 2020. ISSN 1867-2450, 1867-2469. doi: 10.1007/s12551-020-00776-4. URL http://link.springer.com/10.1007/s12551-020-00776-4.

[35] Andrea Carracedo Rodriguez and Serkan Gugercin. The p-AAA algorithm for data driven modeling of parametric dynamical systems. *arXiv:2003.06536 [cs, eess, math]*, March 2020. URL http://arxiv.org/abs/2003.06536. arXiv: 2003.06536.

[36] C. Sanathanan and J. Koerner. Transfer function synthesis as a ratio of two complex polynomials. *IEEE Transactions on Automatic Control*, 8(1):56–58, January 1963. ISSN 0018-9286. doi: 10.1109/TAC.1963.1105517. URL http://ieeexplore.ieee.org/document/1105517/.

[37] Peter J. Schmid. Dynamic Mode Decomposition and Its Variants. *Annual Review of Fluid Mechanics*, 54(1):225–254, January 2022. ISSN 0066-4189, 1545-4479. doi: 10.1146/annurev-fluid-030121-015835. URL https://www.annualreviews.org/doi/10.1146/annurev-fluid-030121-015835.

[38] Adam Semlyen and Bjørn Gustavsen. Rational Approximation of Frequency Domain Responses By Vector Fitting. *IEEE Transactions on Power Delivery*, 14(3):1052–1061, July 1999.

[39] Alvaro Valera-Rivera and Arif Ege Engin. AAA Algorithm for Rational Transfer Function Approximation With Stable Poles. *IEEE Letters on Electromagnetic Compatibility Practice and Applications*, 3(3):92–95, September 2021. ISSN 2637-6423. doi: 10.1109/LEMCPA.2021.3104455. URL https://ieeexplore.ieee.org/document/9512266/.

[40] A. van der Sluis. Condition numbers and equilibration of matrices. *Numerische Mathematik*, 14(1):14–23, December 1969. ISSN 0029-599X, 0945-3245. doi: 10.1007/BF02165096. URL http://link.springer.com/10.1007/BF02165096.

[41] Henk J. van Waarde, Jaap Eising, Harry L. Trentelman, and M. Kanat Camlibel. Data informativity: a new perspective on data-driven analysis and control. *arXiv:1908.00468 [math]*, January 2020. URL http://arxiv.org/abs/1908.00468. arXiv: 1908.00468.

[42] Michel Verhaegen and Patrick Dewilde. Subspace model identification Part 1. The output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210, April 1991. doi: https://doi.org/10.1080/00207179208934363.

[43] Steffen W. R. Werner and Benjamin Peherstorfer. On the sample complexity of stabilizing linear dynamical systems from data. *arXiv:2203.00474 [cs, math]*, February 2022. URL http://arxiv.org/abs/2203.00474. arXiv: 2203.00474.

[44] Kemin Zhou, John Comstock Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, 1996. ISBN 0-13-456567-3.