

RICE UNIVERSITY

**Modeling Systems from Measurements of their Frequency  
Response**

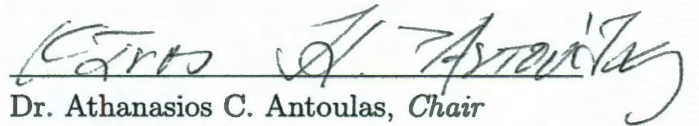
by

**Sanda Lefteriu**

A THESIS SUBMITTED  
IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE

**Doctor of Philosophy**

APPROVED, THESIS COMMITTEE:



Dr. Athanasios C. Antoulas, *Chair*  
Professor  
Electrical and Computer Engineering



Dr. Mark Embree  
Professor  
Computational and Applied Mathematics



Dr. Lin Zhong  
Assistant Professor  
Electrical and Computer Engineering

HOUSTON, TEXAS

MAY 2011

## ABSTRACT

Modeling Systems from Measurements of their Frequency Response

by

Sanda Lefteriu

The problem of modeling systems from frequency response measurements is of interest to many engineers. In electronics, we wish to construct a macromodel from tabulated impedance, admittance or scattering parameters to incorporate it into a circuit simulator for performing circuit analyses. Structural engineers employ frequency response functions to determine the natural frequencies and damping coefficients of the underlying structure. Subspace identification, popular among control engineers, and vector fitting, used by electronics engineers, are examples of algorithms developed for this problem.

This thesis has three goals.

1. For multi-port devices, currently available algorithms are expensive.

This thesis therefore proposes an approach based on the Loewner matrix pencil constructed in the context of tangential interpolation with several possible implementations. They are fast, accurate, build low dimensional models, and are especially designed for a large number of terminals. For noise-free data, they identify the underlying system, rather than merely fitting the measurements. For noisy data, their performance is analyzed for different noise levels introduced in the measurements and an improved version, which identifies an

approximation of the original system even for large noise values, is proposed.

2. This thesis addresses the problem of generating parametric models from measurements performed with respect to the frequency, but also with respect to one or more design parameters, which could relate to geometry or material properties. These models are suited for performing optimization over the design variables. The proposed approach generalizes the Loewner matrix to data depending on two variables.
3. This thesis analyzes the convergence properties of vector fitting, an iterative algorithm that relocates the poles of the model, given some “starting poles” chosen heuristically. It was recognized as a reformulation of the Sanathanan-Koerner iteration and several authors attempted to improve its convergence properties, but a thorough convergence analysis has been missing. Numerical examples show that for high signal to noise ratios, the iteration is convergent, while for low ones, it may diverge. Hence, incorporating a Newton step aims at making the iteration always convergent for “starting poles” chosen close to the solution. A connection between vector fitting and the Loewner framework is exhibited, which resolves the issue of choosing the starting poles.

# Contents

---

<b>Abstract</b>	<b>1</b>
<b>1 Introduction and motivation</b>	<b>3</b>
<b>2 Survey of existing methods and contributions of the thesis</b>	<b>12</b>
2.1 Literature survey . . . . .	12
2.2 Contributions of the thesis . . . . .	16
<b>3 Problem formulation</b>	<b>20</b>
3.1 System theoretic concepts . . . . .	20
3.2 Problem statement . . . . .	23
<b>4 Loewner framework</b>	<b>25</b>
4.1 Theoretical Aspects . . . . .	25
4.1.1 Tangential interpolation . . . . .	29
4.1.2 The Loewner and the shifted Loewner matrices . . . . .	30
4.1.3 The solution to the general tangential interpolation problem in the Loewner framework . . . . .	32
4.1.4 System identification . . . . .	36

---

4.2	Implementation . . . . .	40
4.2.1	Complex approach . . . . .	41
4.2.2	Real approach . . . . .	41
4.2.3	Real alternative approach . . . . .	43
4.2.4	SVD implementation . . . . .	44
4.2.5	Adaptive implementation . . . . .	45
4.2.6	Stability and passivity . . . . .	46
4.2.7	Computational Complexity . . . . .	47
4.2.7.1	Vector fitting . . . . .	47
4.2.7.2	SVD implementation approach . . . . .	47
4.2.7.3	Adaptive implementation approach . . . . .	47
4.2.7.4	Discussion of the algorithms' computational complexity	48
4.3	Noisy measurements . . . . .	48
4.3.1	The Loewner matrix pencil under noisy data . . . . .	48
4.3.2	Proposed Improvements . . . . .	49
4.3.2.1	Noise-free case . . . . .	50
4.3.2.2	$SNR = 80$ . . . . .	51
4.3.2.3	$SNR = 20$ . . . . .	51
4.4	Numerical Results . . . . .	56
4.4.1	Noise-free system with 2 ports, 14 poles and $\mathbf{D} \neq \mathbf{0}$ . . . . .	57
4.4.2	S-parameter measurements . . . . .	59
4.4.3	RC transmission line . . . . .	63
4.4.4	Biochip . . . . .	66
<b>5</b>	<b>Parametric models</b> . . . . .	<b>73</b>
5.1	The Loewner matrix in the two-variable case . . . . .	74
5.2	Generalized state-space realizations . . . . .	77
5.2.1	Single variable realizations . . . . .	77

---

5.2.2	Two variable realizations . . . . .	79
5.3	Obtaining a real realization . . . . .	80
5.4	Generalized state-space realizations for MIMO systems . . . . .	84
5.4.1	One variable case . . . . .	84
5.4.2	Two variable case . . . . .	88
5.5	Numerical examples . . . . .	90
5.5.1	RLC Circuit . . . . .	90
5.5.2	Measured S-parameters . . . . .	92
5.5.3	Convection-Diffusion Equation . . . . .	102
<b>6</b>	<b>Convergence of vector fitting</b>	<b>104</b>
6.1	Review of vector fitting . . . . .	104
6.1.1	Vector fitting for MIMO systems . . . . .	108
6.1.2	Issues with vector fitting . . . . .	110
6.1.3	Vector fitting as a Sanathanan-Koerner iteration . . . . .	112
6.2	Solution to vector fitting using VARPRO . . . . .	113
6.3	Analysis of the pole relocation iteration in vector fitting . . . . .	115
6.4	Least squares solutions . . . . .	117
6.5	Convergence of the pole relocation iteration in vector fitting . . . . .	119
6.6	Incorporating the Newton step in the vector fitting iteration . . . . .	126
6.6.0.1	Recursive Implementation . . . . .	127
6.6.0.2	Implementation with normal equations . . . . .	128
6.6.0.3	Implementation employing the pseudoinverse . . . . .	128
6.7	Proposed reformulation of vector fitting . . . . .	129
6.7.1	Lagrange basis in VF . . . . .	135
6.7.2	Numerical example . . . . .	136
6.7.2.1	Noise-free measurements . . . . .	137
6.7.2.2	Noisy measurements . . . . .	139

---

<b>7 Conclusion</b>	<b>143</b>
<b>References</b>	<b>146</b>
<b>A Proofs</b>	<b>154</b>
A.1 Proof of Theorem 5.4.2 . . . . .	154
<b>B Pseudocodes</b>	<b>158</b>
B.1 SVD implementation in the Loewner framework . . . . .	158
B.2 Adaptive implementation in the Loewner framework . . . . .	158
B.3 Recursive Implementation of the Newton step in the VF iteration . .	160
B.4 Implementation of the Newton step in the VF iteration employing the normal equations . . . . .	160
B.5 Implementation of the Newton step in the VF iteration employing the pseudoinverse . . . . .	161
B.6 Obtaining real quantities in the least squares problem in vector fitting	161
<b>C Additional numerical examples in the Loewner framework</b>	<b>164</b>
C.1 200 S-parameter measurements from a device with 26 ports . . . . .	164
C.2 Delay System . . . . .	168
C.3 Circuit with $p = 70$ ports . . . . .	172
C.4 Example from Sect. C.1 revisited . . . . .	175
<b>D Additional numerical examples analyzing the convergence properties of vector fitting</b>	<b>178</b>
D.1 Order 1 strictly proper system . . . . .	178
D.1.1 2 measurements . . . . .	179
D.1.2 $N \geq 2$ noise-free measurements . . . . .	179
D.1.3 3 noisy measurements . . . . .	181
D.2 Order 2 strictly proper system . . . . .	194

---

D.2.1	5 measurements . . . . .	195
D.3	Numerical example from S-parameter measurements . . . . .	199

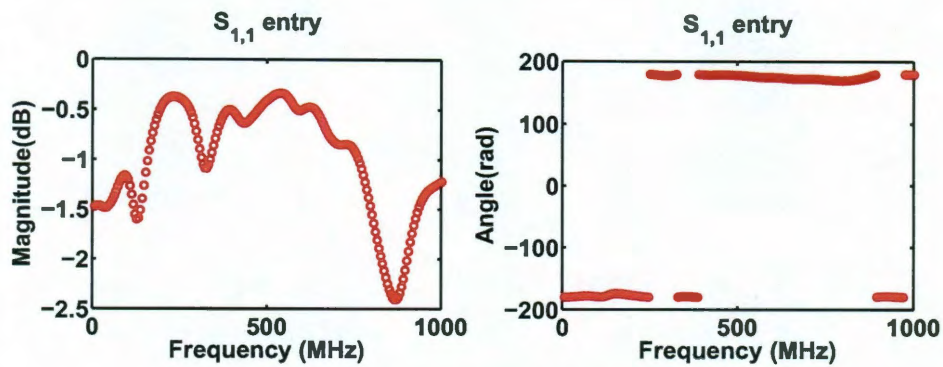


## Introduction and motivation

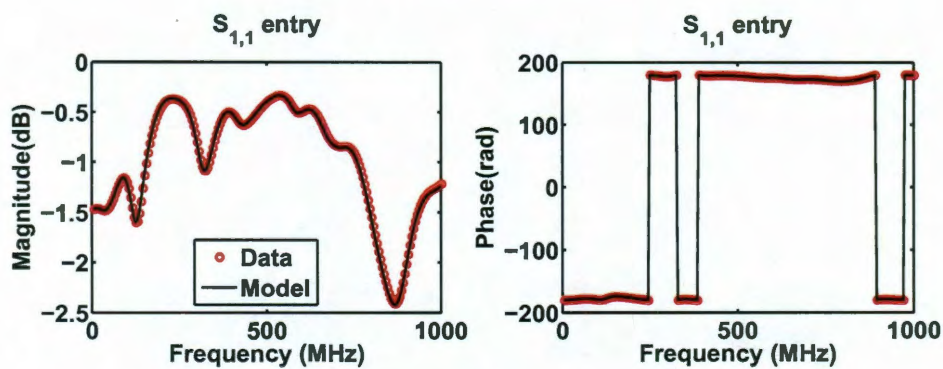
---

The purpose of this introductory chapter is to provide the reader with details regarding the problem treated in the rest of the chapters. As the thesis title indicates, the topics addressed revolve around various aspects of modeling linear dynamical systems from measurements of their frequency response. The statement of the problem at hand is simple: given measurements of a system's frequency response (see Fig. 1.1(a)), one wishes to construct a rational model that matches the data closely (see Fig. 1.1(b)). Having a simple problem statement is by no means equivalent to dealing with an easy problem, as we shall see in the following. Depending on the underlying systems, engineers from various areas have produced an extensive literature on the topic and different algorithms have been proposed (covered partially in Chapter 2).

To obtain the measurements illustrated in Fig. 1.1(a), the following technique is applied. For single-input single-output (SISO) systems, the system is excited with a sinusoidal signal of a certain frequency and the output is observed. At steady-state, the output is a sinusoid of the same frequency, but with a different amplitude and phase. The amplitude change is recoded as the magnitude and the phase change is the phase of a complex number which is used to represent the measurement data for that particular frequency.



(a) Measurement data



(b) Measurement data and the model

Figure 1.1: Modeling a SISO system. Given measurements (here shown as magnitude and phase) of the frequency response of a system (for example, the (1, 1) entry of the scattering parameters of a 26-port device), the goal is to construct a model which matches the data closely.

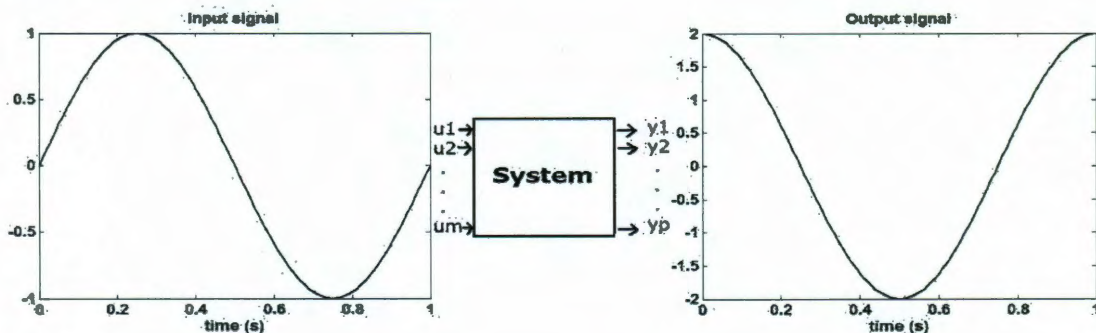


Figure 1.2: Using sinusoidal signals to measure a system's frequency response. The output signal is a sinusoid of the same frequency, but different magnitude and phase.

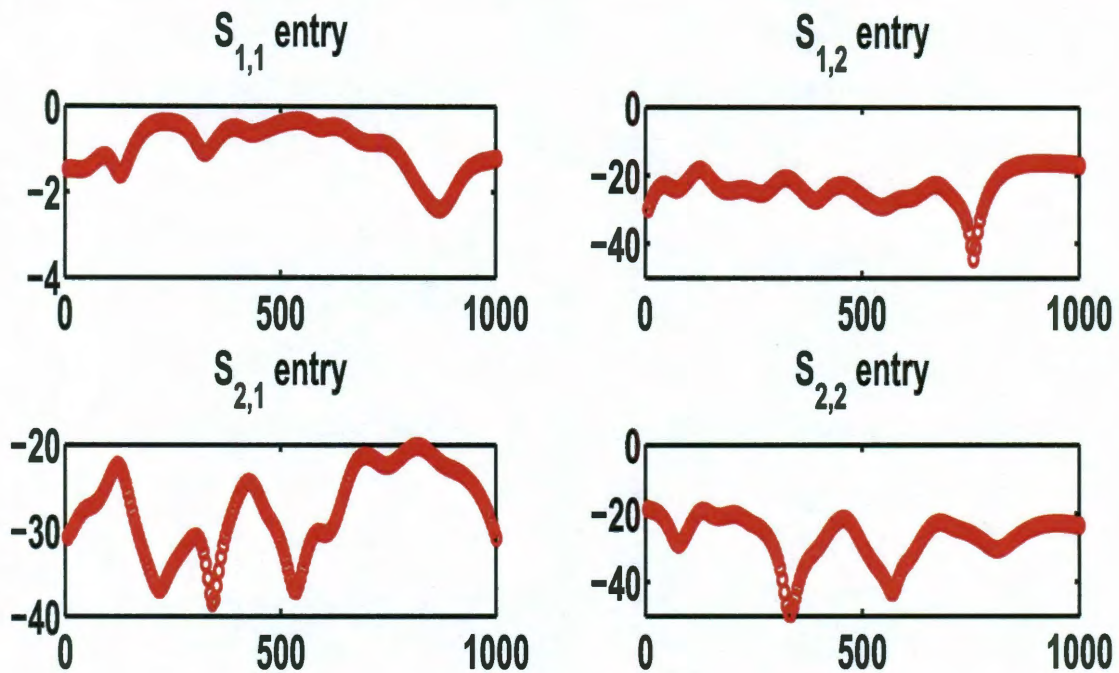


Figure 1.3: Entries of a MIMO system. For MIMO system, the transfer function is a matrix of functions (instead of a scalar, as is the case for SISO systems), and this figure shows different entries of such a system.

In the case of systems with multiple inputs and outputs (MIMO systems), one input is excited and the output is observed at all output ports, so measurement data is provided for each input-output pair. This leads to matrix data: the  $(i, j)^{th}$  entry in the matrix is a complex number describing the magnitude and phase of the output signal applied at input port  $j$  and observed at output port  $i$ . Fig. 1.3 shows the magnitude of different entries of a system with 26 ports (this system is discussed thoroughly in Sect. C.1). Due to the large number of ports, it is unfeasible to check each input-output pair (for a system with  $p = 26$  input and outputs ports, there are  $p^2 = 26^2 = 676$  entries for which the magnitude and phase need to be checked). Therefore, one can examine the  $p$  singular values of the matrix obtained as measurement data for each frequency. Fig. 1.4(a) shows the singular values of the measured frequency response for all ports and Fig. 1.4(b) shows the singular values of the model evaluated for each sample frequency, against those of the data.

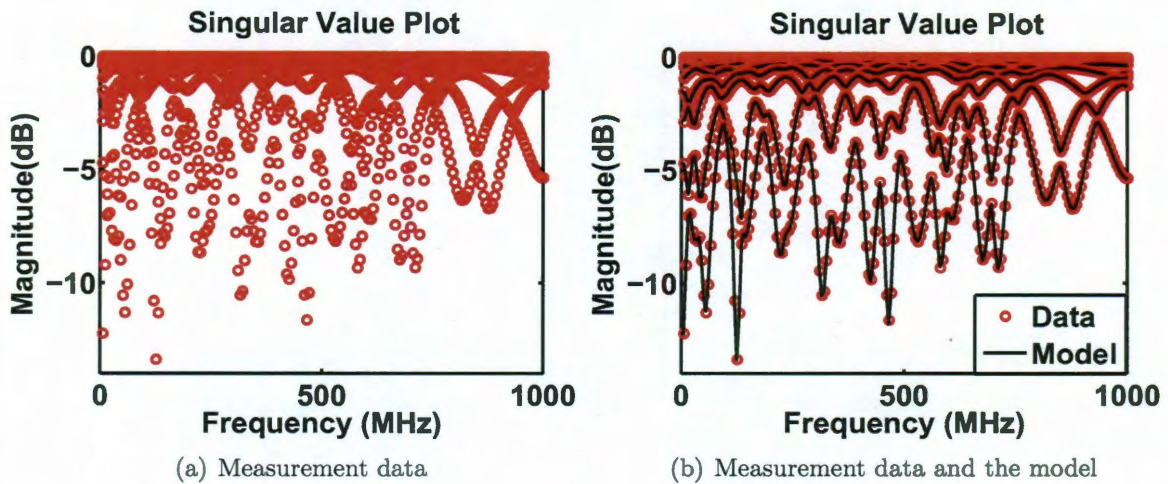


Figure 1.4: Modeling a MIMO system. Instead of checking the accuracy in each entry of the transfer function matrix, it is preferred, especially for multi-port systems, to investigate the accuracy in the singular values of the transfer function matrix.

The problem of building a model from measurements of a system's frequency response is of interest to many engineers. In electronics, the aim is to construct a macromodel from tabulated data representing impedance ( $Z$ -), admittance ( $Y$ -) or scattering ( $S$ -) parameters. When modeling a transmission line as an RLC circuit, one can consider the inputs as the voltage drops at the terminals and the outputs as the currents (as in Fig. 1.5), in which case the admittance transfer function provides frequency response measurements representing the  $Y$ -parameters of the system. However, one can also consider the current injections as inputs and measure the voltage drops, which yields the impedance transfer function. The impedance transfer function is the inverse of the admittance transfer function.

Impedance parameters from one input port are calculated under open circuit conditions for the other ports. Similarly, to measure the admittance parameters from one port, the others are short circuited. On the other hand,  $S$ -parameters use matched loads instead of open or short circuit conditions to characterize a linear network. At high frequencies, matched loads are easier to use, so  $S$ -parameters are preferred to characterize high-frequency behavior. Therefore,  $S$ -parameters are usually measured

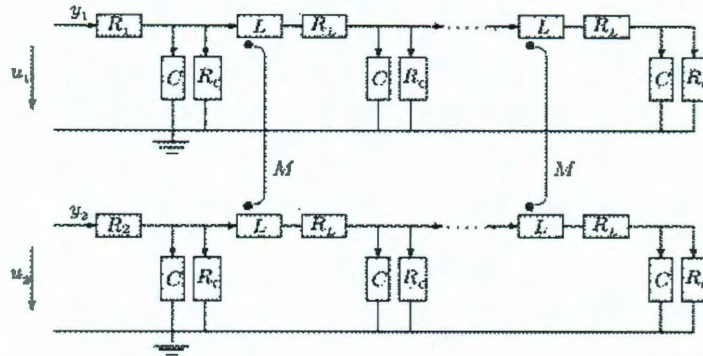


Figure 1.5: Two-port transmission line. Inputs are voltage drops and outputs are currents, hence the measured frequency response provides the admittance parameters.

on wide frequency bands, from low frequencies up to tens of GHz. Scattering parameters are the reflection and transmission coefficients between incident and reflected waves and they are measured by devices called VNA (vector network analyzers). Assuming the same reference impedance  $Z_0$  for all ports (usually  $Z_0 = 50\Omega$ ), these waves may be expressed in terms of the port voltages ( $\mathbf{V}$ ) and currents ( $\mathbf{I}$ ):

$$\mathbf{b} = \frac{1}{2} \frac{\mathbf{V} - Z_0^* \mathbf{I}}{\sqrt{|\operatorname{Re}(Z_0)|}}, \text{ where } \mathbf{b} \text{ are the reflected waves} \quad (1.1)$$

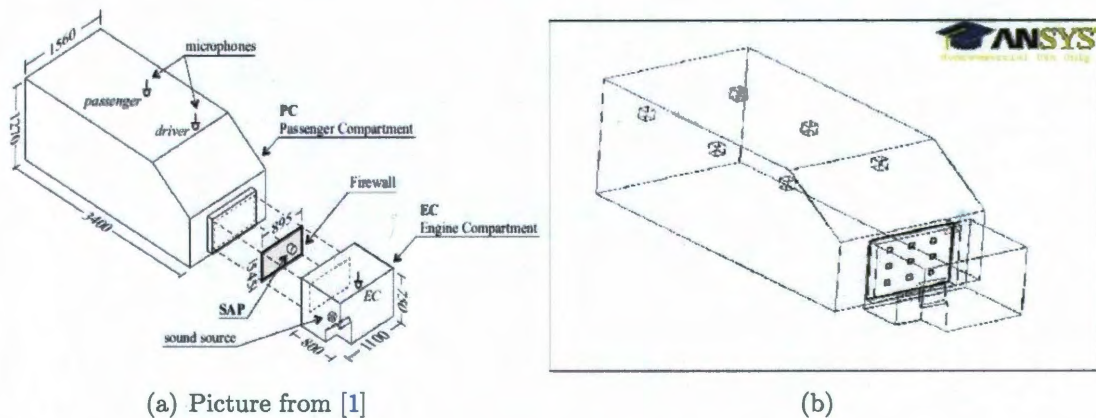
$$\mathbf{a} = \frac{1}{2} \frac{\mathbf{V} + Z_0 \mathbf{I}}{\sqrt{|\operatorname{Re}(Z_0)|}}, \text{ where } \mathbf{a} \text{ are the incident waves.} \quad (1.2)$$

After discussing the different ways one can characterize electrical systems based on the chosen inputs and outputs, this paragraph provides a motivation as to why modeling systems from measurements of their frequency response is an important problem in the electronics industry. Oftentimes, parts are not manufactured in house, and manufacturers only provide to their clients tabulated data that describe the behavior of the electronic component. To integrate the component on the chip and be able to simulate the entire chip, a model is sought for each individual component. Regardless of the kind of tabulated parameters used to characterize the circuit, electronic engineers seek a rational model which should match the tabulated measurements as

closely as possible. This model is subsequently incorporated into a circuit simulator, allowing the engineers to perform various circuit analyses.

For structural engineers, frequency response functions (FRFs) provide an analysis tool for vibrations. Vibration is oftentimes undesirable, as it wastes energy and creates sound (noise). The noise caused by the vibrations of engines in cars or airplanes is unwanted, so careful designs are employed to reduce vibrations. Sound, or “pressure waves”, are generated by vibrating structures, but the latter can also induce the vibration of structures. Hence, reducing noise and vibration are two related problems.

Frequency response functions are measured using a sinusoid forcing function produced by a shaker at some location on the structure. The output is measured as the displacement, velocity or acceleration at one or several points across the structure. FRFs can be employed to build a rational model to determine the natural frequencies and damping coefficients of the underlying system. These allow the engineers to predict when undesired resonance (causing vibration) may occur and to determine the appropriate steps to take to prevent it from occurring.



(a) Picture from [1]

(b)

Figure 1.6: Concrete car

The concrete car (Fig. 1.6(a)) is a test setup mimicking a vehicle. It was built at LMS International as part of the EU FP7-funded RTN project “A computer-aided engineering approach to smart structures” (MRTN-CT-2006-035559) and it contains

two cavities, one as the engine cavity and the other as the passenger cavity, separated by a firewall. There are 9 inputs to the system, namely the piezo-patches attached to the metal frame that separates the two cavities and 6 measured outputs (the sound pressure level at 6 locations in the passenger cavity), so the frequency response function matrix is of dimension  $6 \times 9$  (Fig. 1.6(b)). Fig. 1.7(a) and 1.7(b) show different entries of the FRFs obtained from measurements of the concrete car, as well as of the rational model found using the Loewner framework described in Chapter 4.

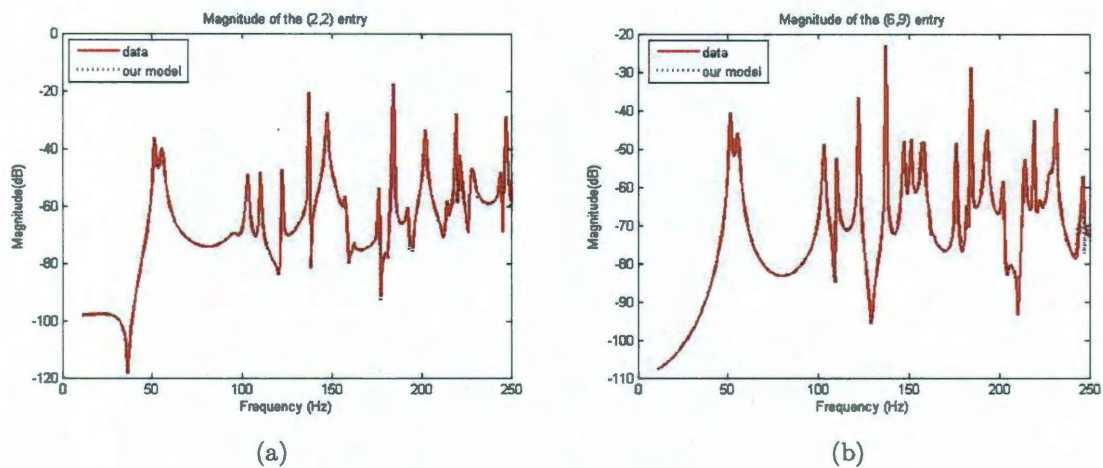


Figure 1.7: Modeling different entries of the transfer function of the concrete car

Another possible application to modeling systems from frequency domain measurements is for large scale systems, for which smaller order models would be obtained by applying model order reduction techniques. Traditionally, model reduction has been applied to systems for which an accurate, but very complex model, is available, obtained, for example, by a finite difference or finite element discretization of the governing equations of the systems, oftentimes expressed as PDEs (e.g., Maxwell's equations for electrical systems, or the heat equation for thermodynamical systems). An alternative technique [2] is model reduction from time domain or frequency domain measurements of devices for which no model is available yet. This black-box technique, in which a reduced order model is built from measurement data, is some-

times preferred by engineers when there is not enough insight available to correctly model the system. Constructing models from time or frequency domain data are essentially different research problems as each presents different challenges, but this thesis focuses on modeling systems from measurements of the frequency response. We do not investigate what makes for a good choice of sampling points, as this is a complex research question by itself. Assuming that the measurements capture the dynamic behavior of interest, chapter 4 presents an efficient framework to construct a macromodel of low complexity that is consistent with the data.

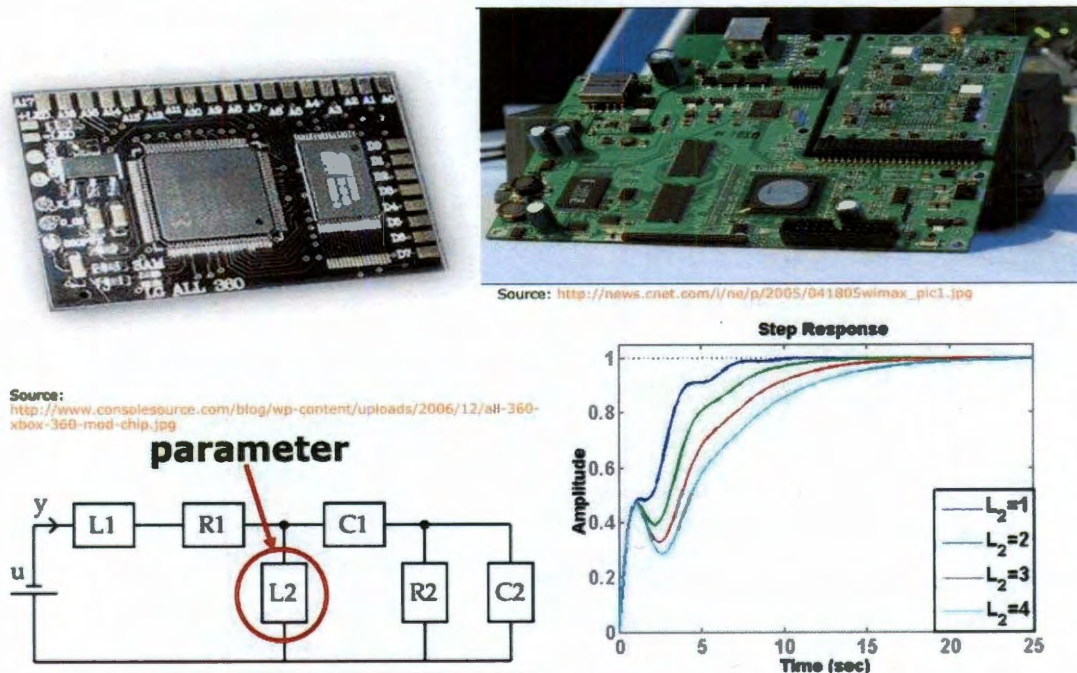


Figure 1.8: Systems depending on parameters. Changing the value of the parameter (the inductor  $L_2$ ) changes the system's response (the step response).

The next chapter, namely chapter 5, investigates the problem of modeling systems from frequency domain measurements in a more general context. Oftentimes, the behavior of dynamical systems (and consequently of their mathematical models) depends on parameters, such as material properties (e.g., dielectric permittivity), geometric characteristics (e.g., spacing between conductors) or varying boundary condi-



tions. Microelectromechanical systems, electronic chip design and VLSI interconnect modeling yield examples of parameterized systems. Fig. 1.8 shows a voltage driven RLC circuit for which the output is measured as the current at that port. The inductor  $L_2$  can be considered as a parameter in the system if its value is not fixed. For different values, the step response shows a different rise time.

The aim is to create *parametric* models that incorporate the dependence on these parameters. This approach is useful in the optimization process arising in product development, since models for many configurations of the design variables are needed for each component to maximize a desired criterion. It is unfeasible to perform new measurements or run a full-wave simulation for each different configuration of the parameter values needed during optimization, and thus having parametric models available ensures a faster design workflow. For fixed parameter values, the frequency-dependent model of, for example, a component in an electrical circuit is rational, so it can be used directly in a circuit simulator.

# Survey of existing methods and contributions of the thesis

---

## 2.1 Literature survey

There is a large body of literature available on the topic of modeling systems from frequency response measurements, which is essentially a rational interpolation problem, and the literature review following in this chapter is by no means exclusive. There are two main approaches. One (e.g., subspace identification, Nevanlinna-Pick interpolation, Loewner framework, etc), in which the model's state-space representation is computed, is based on system theory. The goal of the other techniques (e.g., PolyMAX, the Cauchy method, vector fitting, variable projection, etc.) is fitting the data (i.e., no prior determination of the order of the underlying model), basically interpreting it as an optimization problem in which the deviation between the measurements and the model is minimized. Thus, these procedures are based on least-squares approximations.

Subspace identification, originally developed for time domain measurements [3], is currently implemented in Matlab as the `n4sid` function available in the *System Identification Toolbox*. Together with its frequency-domain version [4], it is widely

used by control engineers.

The algorithm introduced in [5] uses Nevanlinna-Pick interpolation in the context of bounded-real interpolation of S-parameters. As shown in [6], passivity is enforced by interpolating at the original points  $(s_i, \mathbf{H}(s_i))$ , together with the associated *mirror-image* points  $(-s_i^*, \frac{1}{\mathbf{H}(-s_i^*)})$ . Hence this method cannot recover the original system, unless the latter satisfies the mirror-image interpolation constraints, which is generally not the case. Moreover, this work employs matrix interpolation (the entire matrix transfer function is used for each sample), instead of tangential interpolation, which is employed in this thesis. Other algorithms enforcing passivity by construction are described in [7, 8].

PolyMAX [9], a non-iterative method, is mainly used in the automotive and aerospace industries for operational modal analysis. It involves two steps. In the first step, a stabilization diagram is constructed, which shows how the poles of models of different orders evolve when increasing the approximation order. The user needs to select those poles which remain almost constant for all model orders and, based on this selection, the second step consists in determining the mode shapes (i.e., system poles) by solving a second weighted linear least squares problem. The advantage, when compared to frequency domain direct parameter identification (FDPI) [10] or least squares complex exponential (LSCE) [11], the industry-standard, is that the stabilization diagrams are very clear, thus lowering the potential of user mistakes. Just like the least-squares complex frequency-domain (LSCF) estimation method [12], a z-domain model is employed (in other words, a frequency-domain model that is derived from a discrete-time model), so the basis functions used for representing the numerator and denominator of the transfer function are  $\Omega_k(\omega) = e^{j\omega\Delta t k}$  (with  $\Delta t$ , the sampling time).

The Cauchy method [13, 14, 15, 16, 17] is another non-iterative method, in which the numerator and denominator of the transfer function are expressed as polyno-

mials in the monomial basis. A linearization is then employed, which leads to a Vandermonde-type matrix. This matrix is well known to be ill-conditioned for large orders of the polynomials used to model the numerator and denominator of the underlying system transfer function. Moreover, another effect of the ill-conditioning of the Vandermonde matrix occurs when samples span a wide frequency band. To address these issues, authors in [14] proposed the computation of the Vandermonde matrix via an Arnoldi process (however, this approach did not improve the accuracy of the solution), as well as the use of preconditioners and the QR decomposition (these lead to a decrease in the condition number of the Vandermonde matrix, thus yielding accurate solutions). To improve robustness when measurements are corrupted by noise, authors in [17] proposed to solve the overdetermined least squares problem via the total least squares method.

Vector fitting [18, 19, 20, 21, 22, 23, 24, 25, 26] has lately become the method of choice among electronics engineers for constructing a model given tabulated frequency domain data. It was also extended to handle time domain data [27]. This technique solves the rational interpolation problem in two stages: the first for computing the poles of the transfer function and the second for finding the residues. In the first stage, the poles of the model are relocated, using a set of “starting poles” chosen heuristically at the first step, through an iteration process, to their final location. The location of the new poles is determined by solving a least squares problem obtained when linearizing a model for the transfer function of the system with the numerator and denominator expressed in the rational basis. Vector fitting was recognized as a reformulation of the Sanathanan-Koerner iteration [28], but a thorough convergence analysis of the algorithm is still missing. Several authors have attempted to improve its convergence properties. Following the suggestion in [28], authors in [22, 26] developed Orthonormal Vector Fitting, which uses an orthonormal basis obtained via the Gram-Schmidt orthonormalization process applied to the rational basis. Since

the convergence properties of the iteration are impaired in the case of noisy measurements, the improvements proposed in [21] include “skimming” the spurious poles and iteratively increasing the order of the approximant by “adding” new poles.

The authors of [23] performed extensive numerical experiments involving data sets containing 1001 frequency samples, obtained from devices with various port sizes, comparing vector fitting [18] to frequency domain subspace identification [3]. The conclusion was that subspace identification requires a large computational time due to the large number of samples in the data set and, oftentimes, it fails to produce meaningful results. Vector fitting was tested using the element-wise and column-wise implementations for dealing with MIMO systems and, while the method never fails, sometimes the resulting errors are large.

Variable projection [29, 30] can be applied to any problem in which a least squares problem which involves a set of separable linear and nonlinear variables is solved. When expressing the transfer function of the model in pole-residue form, the poles are the nonlinear parameters, while the residues are the linear variables. The linear variables are eliminated and a least squares problem is solved only in terms of the non-linear parameters via a Newton iteration such that the difference between the measurements and the model is minimized at the sampling points.

This thesis also addresses the problem of generating parametric models from measurements performed with respect to the frequency, but also with respect to one or more design parameters, which could relate to geometry or material properties. Several techniques have already been proposed for solving this parametric macromodeling problem. In [31, 32], the authors use vector fitting to first construct models for a set of known parameter values, followed by a parametrization of the numerator and denominator of the transfer function by linear combinations of basis functions that are piecewise linear in each parameter. A generalization of the Sanathanan-Koerner iteration to the parametric case was developed in [33] and improved in [34]. A multivariate

formulation of the Orthonormal Vector Fitting technique was introduced in [35], but the stability of the resulting models was not ensured. Thus, this issue was revisited in [36] by employing barycentric interpolation of univariate nodes. A recursive algorithm to compute the parametrized residues of the multivariate transfer function was presented in [37] and later combined with [35] to provide a hybrid method in [38]. Last, [39] discusses a generalization of multivariate Vector Fitting that includes parameter derivatives. In the related area of parametric model order reduction, the most common approach is interpolatory reduction (see [40, 41, 42], and references therein).

## 2.2 Contributions of the thesis

Current applications require a model of reasonable dimension that approximates the response accurately in the desired frequency range. All available techniques are extremely computationally demanding for constructing low dimensional models for systems with massive ports. The algorithms presented in this thesis employ tangential interpolation in the Loewner matrix pencil framework (see Chapter 4) to address the issue of multi-port systems. Tangential interpolation is of great importance in our context because it allows building reduced models for devices with a large number of input and output ports using a small computational time [43]. Our algorithms are fast and accurate; they construct models of low complexity and are especially designed for devices with many ports, since their computational complexity scales with the second or third power of the number of terminals, as opposed to vector fitting, which scales with the 4<sup>th</sup> power. Also, they allow the identification of the underlying system in the case of enough noise-free measurements. If more measurements are available than needed, plotting the singular values of the Loewner matrix pencil allows the identification of the original system via an SVD truncation. Our algorithms (presented in

Chapter 4) are general and can be applied to any kind of frequency-domain data (see Sect. 4.4 for numerical examples spanning various areas) since we use a black-box approach by not assuming any underlying structure of the systems to be modeled. Moreover, they can construct a non-minimal model exclusively from the available measured data by arranging them in an appropriate way. Another advantage is that, except for the measurement data, no other user input is required. Vector fitting, on the other hand, requires a set of good starting poles for the pole-relocation process to be successful and produce good macromodels, as the outcome of the least-squares optimization depends on the location of the starting poles. The number of starting poles is an additional important parameter for vector fitting which, moreover, requires an additional compacting step [19] for system identification, while in the case of our algorithms, this process is automatic [43].

Modeling systems based on tabulated data is an even more difficult problem because of measurement noise. Due to their limited capability, measurement devices can output only a certain number of digits of the measured quantity, some of which may be wrong. On the other hand, noise should not affect a robust algorithm used to build the model. The effects of noise on our algorithms' performance was investigated and an improvement is proposed in Sect. 4.3.

This thesis addresses the parametric macromodeling problem by generalizing the Loewner matrix to the two variable case in Sect. 5.1 - 5.5. Using the frequency response measurements obtained for several parameter values, we generalize the *Loewner matrix* to the two variable case to construct models that are reduced both with respect to frequency and to the parameter. This generalization is valid for SISO, as well as MIMO systems and it was validated on academic, as well as practical examples consisting of real measurements.

This thesis provides the first analysis ever attempted (to the author's knowledge) to understand the convergence properties of the pole relocation iteration of vector

fitting (in Chapter 6). Experiments show that for high signal to noise ratios, the iteration is convergent, while for low SNRs, the iteration will most likely diverge. Based on this observation, the natural way to proceed is by incorporating a Newton step to guarantee that the iteration will always converge. We illustrate these different situations through simple numerical examples featuring synthetic data.

Sect. 6.7 exhibits a relationship between vector fitting and the Loewner framework [2, 44] that allows us to propose a reformulation meant to address some of the issues in the vector fitting algorithm, namely the choice of starting poles, and determining the true model order (currently, the user guesses the order of the model).

This thesis is organized as follows. Chapter 3 presents a review of system theoretic definitions and basic concepts used in the rest of the chapters. Chapter 4 provides a derivation of the Loewner matrix framework for the scalar case based on Lagrange polynomials, followed by the introduction of the concept of tangential interpolation, and the corresponding Loewner framework for this case. The Loewner framework is completed with the introduction of the shifted Loewner matrix. Sect. 4.1.4 shows that, given the appropriate number of noise-free measurements, the Loewner framework recovers the underlying system. Sect. 4.2 shows various implementation approaches, while Sect. 4.4 presents numerical examples from various areas to validate the proposed implementations. Chapter 5 discusses our approach to solving the parametric macromodeling problem: through a generalization of the Loewner matrix to the two variable case. We introduce the 2D Loewner matrix in Sect. 5.1, while Sect. 5.2 discusses realizations for the single and two variable case in the Loewner framework. Sect. 5.3 presents the left and right basis transformations one needs to perform to obtain a realization containing real coefficients for the case when the measurements provided in one variable come in complex conjugate pairs, while the measurements in the other variable are real. The issue of multiple input multiple output systems is addressed in Sect. 5.4, while Sect. 5.5 concludes this chapter with some numerical



examples. Next, chapter 6 presents an analysis of the convergence properties of the pole relocation iteration of the vector fitting algorithm, as well as a proposed reformulation in Sect. 6.7, a reformulation that is based on insights from the Loewner framework. Lastly, chapter 7 concludes this thesis.

The main contributions of this thesis are:

1. addressing the issue of a large number of ports by employing tangential instead of matrix interpolation in the Loewner framework,
2. addressing the issue of parametric macromodeling by generalizing the Loewner matrix to two-variable transfer functions,
3. providing a convergence analysis of the already existing vector fitting algorithm, as well as several reformulations.

## Problem formulation

---

### 3.1 System theoretic concepts

We present a review of system theoretic definitions and basic concepts which we build upon in the following chapters.

$$\begin{array}{ccc}
 u_1(\cdot) \rightarrow & \boxed{\Sigma : \left\{ \begin{array}{l} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{array} \right.} & \rightarrow y_1(\cdot) \\
 u_2(\cdot) \rightarrow & & \rightarrow y_2(\cdot) \\
 \vdots & & \vdots \\
 u_p(\cdot) \rightarrow & & \rightarrow y_m(\cdot)
 \end{array}$$

**Definition 3.1.1.** *A linear dynamical system  $\Sigma$  with  $m$ -input ports,  $p$ -output ports and  $n$ -internal variables in descriptor-form representation is given by a set of differential and algebraic equations*

$$\Sigma : \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad (3.1)$$

where  $\mathbf{x}(t)$  is an internal variable (the state, if  $\mathbf{E}$  is invertible),  $\mathbf{u}(t)$  is an input,  $\mathbf{y}(t)$  is the corresponding output, while  $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{n \times m}$ ,  $\mathbf{C} \in \mathbb{R}^{p \times n}$ ,  $\mathbf{D} \in \mathbb{R}^{p \times m}$  are constant matrices with  $\mathbf{E}$  possibly singular. Typically,  $\mathbf{E}$  is taken as identity  $\mathbf{I}$ ,

in which case the representation is referred to as the state-space representation of  $\Sigma$ . The first set of equations, namely  $\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$ , describes the dynamics of the system, while the second one simply consists of algebraic equations which define the output variables in terms of the state and input variables.

**Definition 3.1.2.** The set of matrices  $[\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$  is called a realization of  $\mathbf{H}(s)$ . The realization of a transfer function is not unique, but the one of the smallest possible order  $n$  is called a minimal realization. The minimal realization is also not unique, as for any invertible matrices  $\mathbf{T}_1$  and  $\mathbf{T}_2$ , the realization  $[\mathbf{T}_1\mathbf{E}\mathbf{T}_2, \mathbf{T}_1\mathbf{A}\mathbf{T}_2, \mathbf{T}_1\mathbf{B}, \mathbf{C}\mathbf{T}_2, \mathbf{D}]$  corresponds to the same transfer function.

**Definition 3.1.3.** The matrix pencil  $(\mathbf{A}, \mathbf{E})$  is regular if the matrix  $\mathbf{A} - \lambda\mathbf{E}$  is non-singular for some finite  $\lambda \in \mathbb{C}$ .

**Definition 3.1.4.** The poles of a system  $\Sigma$  are the eigenvalues of the pencil  $(\mathbf{A}, \mathbf{E})$ :

$$\text{poles of } \Sigma = p_i = \lambda(\mathbf{A}, \mathbf{E}), i = 1, \dots, n. \quad (3.2)$$

**Definition 3.1.5.**  $\Sigma$  is called stable if all its finite poles are in the left-half plane:

$$\Sigma \text{ stable} \Leftrightarrow \text{Re}(p_i) < 0 \text{ for } |p_i| < \infty, i = 1, \dots, n. \quad (3.3)$$

**Definition 3.1.6.** Several representations are used for the transfer function of  $\Sigma$ :

- $\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ .
- pole-residue form:

$$\mathbf{H}(s) = \sum_{i=1}^n \frac{\mathbf{R}_i}{s - p_i} + \hat{\mathbf{D}},$$

where  $\mathbf{R}_i \in \mathbb{R}^{p \times m}$  are the residue matrices (they are of rank 1) and  $p_i$  are the system poles. Due to the fact that the polynomial term may be incorporated in

the  $\mathbf{E}$  matrix, the  $\mathbf{D}$ -term in the pole-residue expansion  $\hat{\mathbf{D}}$  may be different from the one in the descriptor-form representation  $\mathbf{D}$ .

- pole-zero representation: each entry  $(k, h)$ , for  $k = 1, \dots, p, h = 1, \dots, m$ , of the transfer function is expressed as:

$$\mathbf{H}_{k,h}(s) = \hat{d}_{k,h} \frac{\prod_{i=1}^n (s - z_i^{(k,h)})}{\prod_{i=1}^n (s - p_i)}.$$

The transfer function is proper if  $\lim_{s \rightarrow \infty} \mathbf{H}(s) < \infty$  and improper, otherwise. Moreover, if  $\lim_{s \rightarrow \infty} \mathbf{H}(s) = \mathbf{0}$ ,  $\mathbf{H}(s)$  is strictly proper.

**Definition 3.1.7.** A descriptor system with the pair  $(\mathbf{A}, \mathbf{E})$  regular is completely controllable [45] if  $\text{rank}[\mathbf{A} - \lambda \mathbf{E}, \mathbf{B}] = n$ , for all finite  $\lambda \in \mathbb{C}$  and  $\text{rank}[\mathbf{E}, \mathbf{B}] = n$ . It is called completely observable if  $\text{rank}[\mathbf{A}^T - \lambda \mathbf{E}^T, \mathbf{C}^T] = n$ , for all finite  $\lambda \in \mathbb{C}$  and  $\text{rank}[\mathbf{E}^T, \mathbf{C}^T] = n$ , where  $(\cdot)^T$  denotes transpose.

**Definition 3.1.8.** The  $\mathcal{H}_\infty$  norm of a dynamical system is

$$\|\Sigma\|_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \sigma_1(\mathbf{H}(j\omega)) \quad (3.4)$$

where  $\sigma_1$  is the largest singular value of the transfer function matrix  $\mathbf{H}(s)$  evaluated on the imaginary axis at  $j\omega$ .

In case of a system with only one input and output, the  $\mathcal{H}_\infty$ -norm is the peak on the frequency response. If the system has several inputs and outputs ( $p > 1$ ), the  $\mathcal{H}_\infty$ -norm is given by the maximum of the largest singular value of the  $p \times p$  transfer function matrix  $\mathbf{H}$ , as a function of frequency. Fig. 3.1 illustrates this concept, both for a SISO ( $m = p = 1$ ) system, as well as a MIMO ( $m = p = 2$ ) system.

**Definition 3.1.9.** The  $\mathcal{H}_2$  norm of a dynamical system is

$$\|\Sigma\|_{\mathcal{H}_2} = \left( \frac{1}{2\pi} \int_{-\infty}^{+\infty} \|\mathbf{H}(j\omega)\|^2 d\omega \right)^{\frac{1}{2}}. \quad (3.5)$$

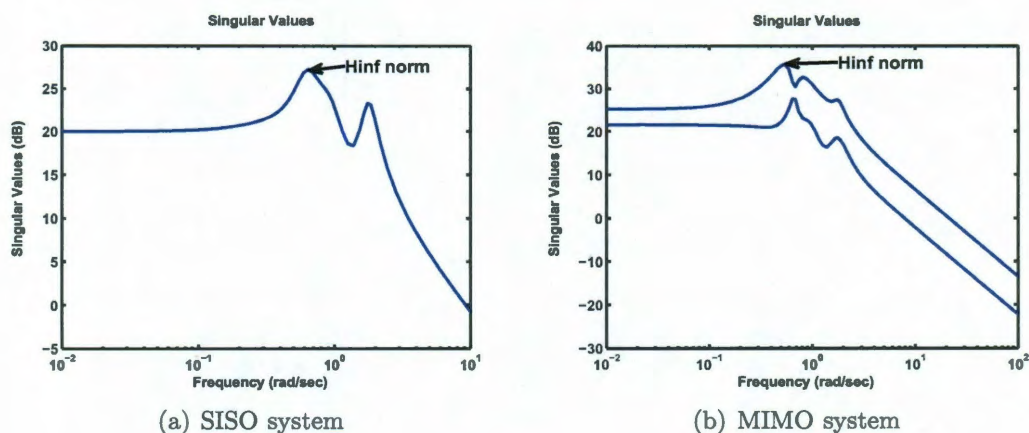


Figure 3.1: Frequency response plots for different systems showing the  $\mathcal{H}_\infty$ -norm

For further details on these issues, we refer to [2].

## 3.2 Problem statement

We say that  $\Sigma$  models the data set obtained from measuring the frequency response of an unknown underlying system with  $m = p$  ports for a number of  $N$  samples

$$\left( j\omega_i, \mathbf{H}^{(i)} := \begin{bmatrix} H_{11}^{(i)} & \dots & H_{1p}^{(i)} \\ \vdots & \vdots & \vdots \\ H_{p1}^{(i)} & \dots & H_{pp}^{(i)} \end{bmatrix} \right), i = 1, \dots, N \quad (3.6)$$

if the value of the transfer function evaluated at the sampling point  $j\omega_i$  is close to the measured matrix at the frequency  $\omega_i$  :

$$\mathbf{H}(j\omega_i) \approx \mathbf{H}^{(i)}, i = 1, \dots, N.$$

The function  $\mathbf{H}(s)$  we are seeking is *rational* (as opposed to a simple polynomial, since transfer functions are rational expressions); this is commonly referred to as the rational interpolation problem. We define the *error matrix* at a specific frequency

which is computed as the difference between the transfer function evaluated at that frequency and the corresponding measurement:

$$\mathbf{H}(j \cdot 2\pi f_i) - \mathbf{H}^{(i)} = \mathbf{Err}(f_i), \quad i = 1, \dots, k. \quad (3.7)$$

If all error matrices have small norm (for instance, in the 2-norm, or their largest singular value), the model is accurate.

## Loewner framework

---

This chapter provides a derivation of the Loewner matrix based on Lagrange polynomials for the scalar case, followed by the introduction of the concept of tangential interpolation and the corresponding Loewner framework for this. The framework is completed by the introduction of the shifted Loewner matrix. These concepts were introduced in [44], while their practical implementation was clarified in [46, 47, 48, 43, 49, 50].

### 4.1 Theoretical Aspects

We start by analyzing the rational interpolation problem in the simple case of scalar data (i.e., one input, one output system, so  $m = p = 1$ ):

$$(s_i, H_i), \quad i = 1, \dots, N, \quad s_i \neq s_j, \quad \text{for } i \neq j, \quad (4.1)$$

where  $s_i, H_i \in \mathbb{C}$  (we use the notation  $H_i$  for the scalar case, instead of  $\mathbf{H}^{(i)}$ , in the general case). The sampling points  $s_i$  are not necessarily on the imaginary axis, as was the case in Eq.(3.7). The rational interpolation problem is equivalent to finding  $\mathbf{H}(s)$  as a ratio of two polynomials, namely  $\frac{n(s)}{d(s)}$ , where  $n, d$  are coprime polynomials, such that  $\mathbf{H}(s_i) = H_i, i = 1, \dots, N$ . Our main tool, the Loewner matrix, is derived

by considering the Lagrange basis for the polynomials  $n(s)$  and  $d(s)$  [51].

We denote by  $\mathcal{P}_k$  the space of all polynomials of degree at most  $k$ . This space has dimension  $k+1$  and one can consider the *monomial basis* (which consists of monomials  $s^i$ ,  $i=0, 1, \dots, k$ ) as a basis for this space. Given  $\lambda_i$ ,  $i=0, \dots, k$ ,  $\lambda_i \neq \lambda_j$ ,  $i \neq j$ , the set of polynomials

$$\mathbf{q}_i(s) := \prod_{i' \neq i} \frac{(s - \lambda_{i'})}{(\lambda_i - \lambda_{i'})}, \quad i = 0, \dots, k, \quad (4.2)$$

yields another basis for the same space  $\mathcal{P}_k$ , the so called *Lagrange basis*. For  $\alpha_i$  set,  $\mathbf{p}(s) := \sum_{i=0}^k \alpha_i \mathbf{q}_i(s)$  is the *Lagrange form of the interpolating polynomial*, which satisfies these  $k+1$  conditions:  $\mathbf{p}(\lambda_i) = \alpha_i$ ,  $i = 0, \dots, k$ .

We can express any rational function  $\mathbf{g}(s)$  as a ratio of two Lagrange polynomials:

$$\mathbf{g}(s) = \frac{\sum_{i=0}^k \beta_i \mathbf{q}_i(s)}{\sum_{i=0}^k \alpha_i \mathbf{q}_i(s)}, \quad (4.3)$$

for arbitrary  $\alpha_i$ ,  $\beta_i$  with  $\alpha_i \neq 0$ .  $\mathbf{g}(s)$  can be completely determined if the  $2(k+1)$  parameters, namely  $\alpha_i$  and  $\beta_i$ , are known. These can be found by setting  $2(k+1)$  interpolation constraints. We first consider  $k+1$  conditions:

$$\mathbf{g}(\lambda_i) = \frac{\beta_i}{\alpha_i} =: \mathbf{w}_i, \quad i = 0, \dots, k. \quad (4.4)$$

By incorporating the constants into the  $\alpha$  and  $\beta$  coefficients and dividing both the numerator and the denominator by  $\prod_{i=1}^k (s - \lambda_i)$ , we can rewrite  $\mathbf{g}(s)$  as

$$\mathbf{g}(s) = \frac{\sum_{i=0}^k \frac{\beta_i}{s - \lambda_i}}{\sum_{i=0}^k \frac{\alpha_i}{s - \lambda_i}}. \quad (4.5)$$

By multiplying with the denominator, bringing everything on one side and using Eq.(4.4) to group terms, we obtain that  $\mathbf{g}$  satisfies the equation

$$\sum_{i=0}^k \alpha_i \frac{\mathbf{g} - \mathbf{w}_i}{s - \lambda_i} = 0, \quad \alpha_i \neq 0. \quad (4.6)$$

We determine  $\alpha_i$  (and consequently,  $\beta_i$ , from  $\beta_i = \mathbf{w}_i \alpha_i$  in Eq.(4.4), by considering



$h + 1$  ( $h \geq k$ ) additional interpolation conditions:

$$\mathbf{g}(\mu_j) = \mathbf{v}_j, j=0, \dots, h, \quad (4.7)$$

where  $(\mu_j, \mathbf{v}_j)$  are pairs of complex numbers ( $\mu_i \neq \mu_j$ , for  $i \neq j$  and  $\lambda_i \neq \mu_j$  for  $\forall i, j$ ). Substituting (4.7) in (4.6) we obtain the following condition for  $\alpha_i$ :  $\mathbb{L}\mathbf{c} = 0$ , where

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{v}_0 - \mathbf{w}_0}{\mu_0 - \lambda_0} & \dots & \frac{\mathbf{v}_0 - \mathbf{w}_k}{\mu_0 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_h - \mathbf{w}_0}{\mu_h - \lambda_0} & \dots & \frac{\mathbf{v}_h - \mathbf{w}_k}{\mu_h - \lambda_k} \end{bmatrix} \in \mathbb{C}^{(h+1) \times (k+1)}, \quad \mathbf{c} = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_k \end{bmatrix} \in \mathbb{C}^{k+1}. \quad (4.8)$$

The matrix  $\mathbb{L}$  is the Loewner matrix [52, 53, 44, 2] associated with the partition of  $(s_i, H_i)$  into the *row array*  $(\mu_j, \mathbf{v}_j)$ , and the *column array*  $(\lambda_i, \mathbf{w}_i)$ . It was first used for solving rational interpolation problems in [52]. Thus  $\mathbf{g}$  is found by determining  $\mathbf{c}$  in the right kernel of  $\mathbb{L}$ .

One such partition could be  $(\lambda_i = s_i, w_i = H_i)$ ,  $i = 1, \dots, \frac{N}{2}$  and  $(\mu_j = s_{j+\frac{N}{2}}, v_j = H_{j+\frac{N}{2}})$ ,  $j = 1, \dots, \frac{N}{2}$ , if  $N$  is even, or,  $(\lambda_i = s_i, w_i = H_i)$ ,  $i = 1, \dots, \frac{N+1}{2}$  and  $(\mu_j = s_{j+\frac{N+1}{2}}, v_j = H_{j+\frac{N+1}{2}})$ ,  $j = 1, \dots, \frac{N-1}{2}$ , if  $N$  is odd. When dealing with noise-free measurements, the chosen partition of the measurements will not influence the results. However, for noisy data, results will be different depending on the way measurements are partitioned into the row and column arrays. This is due to the fact that interpolation at the sampling points in the column array is exact (these make up the Lagrange polynomials in Eq.(4.2)), while interpolation at the sampling points in the row array is approximate (these determine the weights of the Lagrange polynomials).

There are several reasons why this is the right tool to use. The rank of the Loewner matrix constructed from noise-free measurements using any partition is the same as the degree of the minimal interpolant of the data, defined as the maximum between the degrees of the polynomials in the numerator and denominator. This property is

due to its system theoretic interpretation in terms of the generalized observability matrix,  $\mathcal{O}$ , the underlying  $\mathbf{E}$  matrix and the generalized controllability matrix,  $\mathcal{R}$ . In particular, suppose that the data is obtained by sampling a strictly proper rational function  $\mathbf{H}(s)$  with minimal state space representation  $[\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{0}]$ . Then

$$\mathbb{L} = - \underbrace{\begin{bmatrix} \mathbf{C}(\lambda_1 \mathbf{E} - \mathbf{A})^{-1} \\ \vdots \\ \mathbf{C}(\lambda_k \mathbf{E} - \mathbf{A})^{-1} \end{bmatrix}}_{\mathcal{O}} \mathbf{E} \underbrace{\begin{bmatrix} (\mu_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} & \dots & (\mu_h \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \end{bmatrix}}_{\mathcal{R}}.$$

Assuming a strictly proper system, the matrix  $\mathbf{E}$  is full rank. The system is completely controllable and observable, so the controllability and observability matrices  $\mathcal{R}$  and  $\mathcal{O}$ , respectively, are also full rank. Therefore, the rank of  $\mathbb{L}$  is precisely the rank of  $\mathbf{E}$ , which is the order of the minimal degree interpolant of the data [54]. This factorization of the Loewner matrix was used to arrive at a state-space realization in [53]. Last, for data given at a single point with multiplicity:  $(s_1; H_1, H_1^{(1)}, \dots, H_1^{(N-1)})$ , i.e. the value of the function at  $s_1$  and that of  $N-1$  of its derivatives, the Loewner matrix has Hankel structure, so the Loewner matrix generalizes the Hankel matrix [52, 55, 53].

**Example 4.1.1.** Consider this simple example. The transfer function generating the data is  $\mathbf{H}(s) = \frac{1}{s+5}$  and we are given  $N = 4$  samples of this function:

$$(s, H) = \left\{ \left(1, \frac{1}{6}\right), \left(2, \frac{1}{7}\right), \left(-1, \frac{1}{4}\right), \left(-2, \frac{1}{3}\right) \right\} \quad (4.9)$$

We partition this set as follows:

$$k = 1, (\lambda, w) = \left\{ \left(1, \frac{1}{6}\right), \left(2, \frac{1}{7}\right) \right\}, \quad h = 1, (\mu, v) = \left\{ \left(-1, \frac{1}{4}\right), \left(-2, \frac{1}{3}\right) \right\}, \quad (4.10)$$

and build the Loewner matrix according to Eq.(4.8):

$$\mathbb{L} = \begin{bmatrix} -\frac{1}{24} & -\frac{1}{28} \\ -\frac{1}{18} & -\frac{1}{21} \end{bmatrix} \in \mathbb{C}^{2 \times 2}. \quad (4.11)$$

One can easily see that the matrix above is singular, as its determinant is 0. Thus, it follows that the rank of  $\mathbb{L}$  is 1, so the interpolant for these 4 points has minimal degree 1, which is indeed the case, since  $\mathbf{H}(s) = \frac{1}{s+5}$  is of degree 1. The null space of the Loewner matrix is  $\mathbf{c}^T = \begin{bmatrix} 6 & -7 \end{bmatrix}$ , so  $\alpha_0 = 6$ ,  $\alpha_1 = -7$ ,  $\beta_0 = 6 \cdot \frac{1}{6} = 1$ ,  $\beta_1 = -7 \cdot \frac{1}{7} = -1$ . Therefore, the interpolant is as expected:

$$\mathbf{g}(s) = \frac{\frac{1}{s-1} - \frac{1}{s-2}}{\frac{6}{s-1} - \frac{7}{s-2}} = \frac{1}{s+5} \quad (4.12)$$

### 4.1.1 Tangential interpolation

In the case in which a rational matrix function is sought, tangential interpolation is of interest [44]. This concept was investigated in [56] from the model reduction point of view. The data consist now of the *right interpolation data*, denoted as

$$\{(\lambda_i, \mathbf{r}_i, \mathbf{w}_i) \mid \lambda_i \in \mathbb{C}, \mathbf{r}_i \in \mathbb{C}^{m \times 1}, \mathbf{w}_i \in \mathbb{C}^{p \times 1}\}, \quad (4.13)$$

for  $i = 1, \dots, k$ , or, more compactly,

$$\Lambda = \text{diag} [\lambda_1, \dots, \lambda_k] \in \mathbb{C}^{k \times k}, \quad (4.14)$$

$$\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_k] \in \mathbb{C}^{m \times k}, \quad (4.15)$$

$$\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k] \in \mathbb{C}^{p \times k}, \quad (4.16)$$

and of the *left interpolation data*

$$\{(\mu_j, \ell_j, \mathbf{v}_j) \mid \mu_j \in \mathbb{C}, \ell_j \in \mathbb{C}^{1 \times p}, \mathbf{v}_j \in \mathbb{C}^{1 \times m}\}, \quad (4.17)$$

for  $j = 1, \dots, h$ , or, more compactly,

$$M = \text{diag} [\mu_1, \dots, \mu_h] \in \mathbb{C}^{h \times h}, \quad (4.18)$$

$$\mathbf{L} = \begin{bmatrix} \ell_1 \\ \vdots \\ \ell_h \end{bmatrix} \in \mathbb{C}^{h \times p}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_h \end{bmatrix} \in \mathbb{C}^{h \times m}. \quad (4.19)$$

The quantities  $\lambda_i, \mu_j$  are points at which the function is evaluated;  $\mathbf{r}_i, \ell_j$  are referred to as tangential directions on the right and on the left; and  $\mathbf{w}_i, \mathbf{v}_j$  are right and left tangential data, respectively.

Tangential data may be given as above, but most often matrix data, i.e., the value of a transfer function matrix at several points, is provided. Modeling measured S-parameters puts us in the last scenario, as we are given data as in Eq.(3.6), so tangential data can be obtained by following Sect. 4.2.1.

The rational interpolation problem for tangential data aims at finding a realization  $[\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ , such that the associated transfer function satisfies the *right and left interpolation conditions*

$$\mathbf{H}(\lambda_i)\mathbf{r}_i = \mathbf{w}_i, \quad \ell_j\mathbf{H}(\mu_j) = \mathbf{v}_j. \quad (4.20)$$

The key tools for studying this problem are the *Loewner matrix*, variously called the *divided-difference matrix* and *null-pole coupling matrix*, together with the *shifted Loewner matrix* associated with the data; for the material in Sect. 4.1.2 and 4.1.3, we refer to [44] for details on proofs and derivations.

#### 4.1.2 The Loewner and the shifted Loewner matrices

Given a set  $S = \{s_1, \dots, s_N\}$  of points in the complex plane and the evaluation of a rational matrix function  $\mathbf{H}(s)$  at those points:  $\{\mathbf{H}(s_1), \dots, \mathbf{H}(s_N)\}$ , we partition  $S$ :

$$S = \{\lambda_1, \dots, \lambda_k\} \cup \{\mu_1, \dots, \mu_h\},$$

where  $k + h = N$ . Matrix data leads to tangential data by selecting directions  $\mathbf{r}_i$  and  $\ell_j$  (see Sect. 4.2). The Loewner matrix using tangential data is built as:

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{v}_1 \mathbf{r}_1 - \ell_1 \mathbf{w}_1}{\mu_1 - \lambda_1} & \dots & \frac{\mathbf{v}_1 \mathbf{r}_k - \ell_1 \mathbf{w}_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_h \mathbf{r}_1 - \ell_h \mathbf{w}_1}{\mu_h - \lambda_1} & \dots & \frac{\mathbf{v}_h \mathbf{r}_k - \ell_h \mathbf{w}_k}{\mu_h - \lambda_k} \end{bmatrix}. \quad (4.21)$$

It satisfies the following Sylvester equation

$$\mathbb{L}\Lambda - M\mathbb{L} = \mathbf{L}\mathbf{W} - \mathbf{V}\mathbf{R} \quad (4.22)$$

and has a system theoretic interpretation, this time in terms of the tangential controllability and observability matrices. For strictly proper systems, we could use, for example, the SVD to factor  $\mathbb{L}$  and identify a realization [53]. If the system has a nonzero or hidden  $\mathbf{D}$ -term, identification is more involved. This motivates introducing

$$\sigma\mathbb{L} = \begin{bmatrix} \frac{\mu_1 \mathbf{v}_1 \mathbf{r}_1 - \lambda_1 \ell_1 \mathbf{w}_1}{\mu_1 - \lambda_1} & \dots & \frac{\mu_1 \mathbf{v}_1 \mathbf{r}_k - \lambda_k \ell_1 \mathbf{w}_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mu_h \mathbf{v}_h \mathbf{r}_1 - \lambda_1 \ell_h \mathbf{w}_1}{\mu_h - \lambda_1} & \dots & \frac{\mu_h \mathbf{v}_h \mathbf{r}_k - \lambda_k \ell_h \mathbf{w}_k}{\mu_h - \lambda_k} \end{bmatrix}, \quad (4.23)$$

called the *shifted Loewner matrix*, which is the Loewner matrix corresponding to  $s\mathbf{H}(s)$ . It also satisfies a Sylvester equation

$$\sigma\mathbb{L}\Lambda - M\sigma\mathbb{L} = \mathbf{L}\mathbf{W}\Lambda - M\mathbf{V}\mathbf{R}. \quad (4.24)$$

These Sylvester equations allow for an efficient Matlab implementation to initialize the matrices, by avoiding nested for loops to initialize entries one by one [43].

### 4.1.3 The solution to the general tangential interpolation problem in the Loewner framework

Next we review the conditions for the solution of the general tangential interpolation problem by means of state-space matrices  $[\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ , as presented in [44].

**Theorem 4.1.1.** *Assume that  $k = h$  and that  $\det(x\mathbb{L} - \sigma\mathbb{L}) \neq 0$ , for all  $x \in \{\lambda_i\} \cup \{\mu_j\}$  (i.e., the matrix pencil  $(\sigma\mathbb{L}, \mathbb{L})$  is regular and  $\mu_j, \lambda_i \notin \lambda(\sigma\mathbb{L}, \mathbb{L})$ ). Then  $\mathbf{E} = -\mathbb{L}$ ,  $\mathbf{A} = -\sigma\mathbb{L}$ ,  $\mathbf{B} = \mathbf{V}$ ,  $\mathbf{C} = \mathbf{W}$  and  $\mathbf{D} = \mathbf{0}$  is a minimal realization of an interpolant of the data. Thus, the associated transfer function*

$$\mathbf{H}(s) = \mathbf{W}(\sigma\mathbb{L} - s\mathbb{L})^{-1}\mathbf{V} \quad (4.25)$$

satisfies the left and right interpolation conditions:  $\ell_j\mathbf{H}(\mu_j) = \mathbf{v}_j$ ,  $\mathbf{H}(\lambda_i)\mathbf{r}_i = \mathbf{w}_i$ .

This theorem holds if the number of samples are not more than needed ( $k$  and  $h$  are smaller or equal to  $n + \text{rank}(\mathbf{D})$ , where  $n$  is the McMillan degree of the system and the  $\text{rank}(\mathbf{D})$  term comes from the fact that the realization above has a  $\mathbf{D} = \mathbf{0}$  term). If  $k$  and  $h$  are larger than  $n + \text{rank}(\mathbf{D})$  and measurements are noise-free, the assumption that the matrix pencil  $(\sigma\mathbb{L}, \mathbb{L})$  is regular does not hold.

**Example 4.1.2.** *We revisit the previous example from the point of view of tangential interpolation. This was also discussed in [50]. We consider the transfer function  $\mathbf{H}(s) = \frac{1}{s+5}$ , as well as right tangential data  $(\lambda, \mathbf{r}, \mathbf{w}) = (1, 1, \frac{1}{6})$  (i.e.,  $k = 1$ ) and left tangential data  $(\mu, \ell, \mathbf{v}) = (-1, 1, \frac{1}{4})$  (i.e.,  $h = 1$ ). Compactly, we write this as*

$$\Lambda = \begin{bmatrix} 1 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 1 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} \frac{1}{6} \end{bmatrix} \text{ and } M = \begin{bmatrix} -1 \end{bmatrix}, \mathbf{L} = \begin{bmatrix} 1 \end{bmatrix}, \mathbf{V} = \begin{bmatrix} \frac{1}{4} \end{bmatrix}.$$

The Loewner matrix pencil is built according to Eq.(4.21) and (4.23):

$$\mathbb{L} = \begin{bmatrix} -\frac{1}{24} \end{bmatrix} \in \mathbb{C}^{1 \times 1}, \quad \sigma\mathbb{L} = \begin{bmatrix} \frac{5}{24} \end{bmatrix} \in \mathbb{C}^{1 \times 1}.$$

The theorem's assumptions are satisfied:  $k = h = 1$ ,  $\det(x\mathbb{L} - \sigma\mathbb{L}) = \det(x(-\frac{1}{24}) - \frac{5}{24}) \neq 0$

0, for  $x \in \{1, -1\}$  and  $\mu_j = -1, \lambda_i = 1 \notin \text{eig}(\sigma\mathbb{L}, \mathbb{L})$ , so we directly write the realization

$$\mathbf{E}_r = \frac{1}{24}, \mathbf{A}_r = -\frac{5}{24}, \mathbf{B}_r = \frac{1}{4}, \mathbf{C}_r = \frac{1}{6}, \mathbf{D} = \mathbf{0} \Rightarrow \mathbf{H}(s) = \frac{1}{6} \left( s \frac{1}{24} + \frac{5}{24} \right)^{-1} \frac{1}{4}.$$

The transfer function associated to this realization is the same as the original  $\frac{1}{s+5}$ .

**Example 4.1.3.** We discuss an example that shows how to apply the Loewner framework to modeling systems from measurements of their frequency response. This example was also considered in [50]. The transfer function generating the data is

$$\mathbf{H}(s) = \frac{1}{s^2 + 5s + 6} \begin{bmatrix} s & -6 \\ 1 & s + 5 \end{bmatrix}. \quad (4.26)$$

As tangential interpolation data we have

$$k = 2, (\lambda, \mathbf{r}, \mathbf{w}) = \left\{ \left( i, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \frac{1}{5+5i} \begin{bmatrix} i \\ 1 \end{bmatrix} \right), \left( 2i, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \frac{1}{2+10i} \begin{bmatrix} -6 \\ 5+2i \end{bmatrix} \right) \right\}, \quad (4.27)$$

$$h = 2, (\mu, \ell, \mathbf{v}) = \left\{ \left( -i, \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T, \frac{-1}{5-5i} \begin{bmatrix} i \\ 6 \end{bmatrix}^T \right), \left( -2i, \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T, \frac{1}{2-10i} \begin{bmatrix} 1 \\ 5-2i \end{bmatrix}^T \right) \right\}. \quad (4.28)$$

Tangential data leads to the following matrices

$$\Lambda = \begin{pmatrix} i & \\ & 2i \end{pmatrix}, M = \begin{pmatrix} -i & \\ & -2i \end{pmatrix}, \mathbf{R} = \mathbf{L} = \mathbf{I}_2,$$

$$\mathbf{W} = \begin{bmatrix} \frac{i}{5+5i} & \frac{-6}{2+10i} \\ \frac{1}{5+5i} & \frac{5+2i}{2+10i} \end{bmatrix}, \mathbf{V} = \begin{bmatrix} \frac{-i}{5-5i} & \frac{-6}{5-5i} \\ \frac{1}{2-10i} & \frac{5-2i}{2-10i} \end{bmatrix}$$

and the Loewner matrix pencil is given by

$$\mathbb{L} = \begin{bmatrix} \frac{1}{10} & \frac{51}{130} - \frac{21}{130}i \\ -\frac{17}{260} - \frac{7}{260}i & -\frac{23}{104} \end{bmatrix}, \quad \sigma\mathbb{L} = \begin{bmatrix} \frac{1}{10} & -\frac{18}{65} + \frac{12}{65}i \\ \frac{3}{65} + \frac{2}{65}i & \frac{15}{52} \end{bmatrix}. \quad (4.29)$$

Since the theorem's assumptions are satisfied, we can write a realization as

$$\mathbf{E}_r = - \begin{bmatrix} \frac{1}{10} & \frac{51}{130} - \frac{21}{130}i \\ -\frac{17}{260} - \frac{7}{260}i & -\frac{23}{104} \end{bmatrix}, \quad \mathbf{A}_r = - \begin{bmatrix} \frac{1}{10} & -\frac{18}{65} + \frac{12}{65}i \\ \frac{3}{65} + \frac{2}{65}i & \frac{15}{52} \end{bmatrix}, \quad (4.30)$$

$$\mathbf{B}_r = \begin{bmatrix} \frac{-i}{5-5i} & \frac{-6}{5-5i} \\ \frac{1}{2-10i} & \frac{5-2i}{2-10i} \end{bmatrix}, \quad \mathbf{C}_r = \begin{bmatrix} \frac{i}{5+5i} & \frac{-6}{2+10i} \\ \frac{1}{5+5i} & \frac{5+2i}{2+10i} \end{bmatrix}. \quad (4.31)$$

The transfer function is as expected:

$$\mathbf{H}_r(s) = \mathbf{C}_r (s\mathbf{E}_r - \mathbf{A}_r)^{-1} \mathbf{B}_r = \frac{1}{s^2 + 5s + 6} \begin{bmatrix} s & -6 \\ 1 & s + 5 \end{bmatrix}. \quad (4.32)$$

The following remarks are in place:

- Instead of the matrix obtained by evaluating the transfer function at  $s = i$ ,

$$\mathbf{H}(i) = \begin{bmatrix} \frac{i}{i^2+5i+6} & \frac{-6}{i^2+5i+6} \\ \frac{1}{i^2+5i+6} & \frac{i+5}{i^2+5i+6} \end{bmatrix} = \begin{bmatrix} \frac{i}{5i+5} & \frac{-6}{5i+5} \\ \frac{1}{5i+5} & \frac{i+5}{5i+5} \end{bmatrix}, \quad (4.33)$$

which would be used in matrix interpolation, tangential data uses only the vector  $\mathbf{w}_1 = \mathbf{H}(i)\mathbf{r}_1$ , in our case the first column of  $\mathbf{H}(i)$ , since  $\mathbf{r}_1$  is the first identity vector. Therefore, instead of interpolating the entire matrix, we interpolate it along a direction  $\mathbf{w}_1$ , which is obtained by multiplying the matrix measurement by a constant vector  $\mathbf{r}_1$ . In our example,  $\mathbf{r}_1$ , the first identity vector, selects the first column of  $\mathbf{H}(i)$ , yielding  $\mathbf{w}_1$ . In general,  $\mathbf{w}_1$  is a linear combination of the columns of the matrix measurement  $\mathbf{H}(i)$ . A similar observation holds for  $\mathbf{w}_2$ . The left data  $\mathbf{v}_1, \mathbf{v}_2$  are, in general, linear combinations of the rows of the matrix measurement at the sampling points  $\mu_1, \mu_2$ .



- In the realization  $[\mathbf{E}_r, \mathbf{A}_r, \mathbf{B}_r, \mathbf{C}_r, \mathbf{D}_r]$ , the matrices have complex entries, so using the same matrix data and considering the tangential data as

$$k = 2, (\lambda, \mathbf{r}, \mathbf{w}) = \left\{ \left( i, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \frac{1}{5+5i} \begin{bmatrix} i \\ 1 \end{bmatrix} \right), \left( -i, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \frac{1}{5-5i} \begin{bmatrix} -i \\ 1 \end{bmatrix} \right) \right\}, \quad (4.34)$$

$$h = 2, (\mu, \ell, \mathbf{v}) = \left\{ \left( 2i, \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T, \frac{1}{2+10i} \begin{bmatrix} 2i \\ -6 \end{bmatrix}^T \right), \left( -2i, \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T, \frac{1}{2-10i} \begin{bmatrix} -2i \\ -6 \end{bmatrix}^T \right) \right\} \quad (4.35)$$

(i.e., points and their conjugates on the same side) leads to the realization

$$\begin{aligned} \mathbf{E}_r = -\mathbf{L} = - \begin{bmatrix} -\frac{4}{65} - \frac{6}{65}i & \frac{3}{65} - \frac{2}{65}i \\ \frac{3}{65} + \frac{2}{65}i & -\frac{4}{65} + \frac{6}{65}i \end{bmatrix}, \quad \mathbf{A}_r = -\sigma\mathbf{L} = - \begin{bmatrix} \frac{37}{130} - \frac{3}{130}i & \frac{21}{130} - \frac{1}{130}i \\ \frac{21}{130} + \frac{1}{130}i & \frac{37}{130} + \frac{3}{130}i \end{bmatrix}, \\ \mathbf{B}_r = \mathbf{V} = \begin{bmatrix} \frac{5}{26} + \frac{1}{26}i & -\frac{3}{26} + \frac{15}{26}i \\ \frac{5}{26} - \frac{1}{26}i & -\frac{3}{26} - \frac{15}{26}i \end{bmatrix}, \quad \mathbf{C}_r = \mathbf{W} = \begin{bmatrix} \frac{1}{10} + \frac{1}{10}i & \frac{1}{10} - \frac{1}{10}i \\ \frac{1}{10} - \frac{1}{10}i & \frac{1}{10} + \frac{1}{10}i \end{bmatrix}, \quad \mathbf{D}_r = \mathbf{0}_2. \end{aligned}$$

After applying the coordinate change  $\Pi = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}$ , a realization with real entries is obtained as:

$$\begin{aligned} \mathbf{E}_n = -\Pi^*\mathbf{L}\Pi = - \begin{bmatrix} -\frac{1}{65} & -\frac{4}{65} \\ \frac{8}{65} & -\frac{7}{65} \end{bmatrix}, \quad \mathbf{A}_n = -\Pi^*\sigma\mathbf{L}\Pi = - \begin{bmatrix} \frac{29}{65} & -\frac{1}{65} \\ \frac{2}{65} & \frac{8}{65} \end{bmatrix}, \\ \mathbf{B}_n = \Pi^*\mathbf{V} = \begin{bmatrix} \frac{486}{1787} & -\frac{460}{2819} \\ -\frac{473}{8696} & -\frac{1458}{1787} \end{bmatrix}, \quad \mathbf{C}_n = \mathbf{W}\Pi = \begin{bmatrix} \frac{197}{1393} & \frac{197}{1393} \\ \frac{197}{1393} & -\frac{197}{1393} \end{bmatrix}, \quad \mathbf{D}_n = \mathbf{0}_2. \end{aligned}$$

The associated transfer function is as in Eq.(4.32), as expected.

- Tangential directions in Eq.(4.34)-(4.35) are equal, but this still allows us to identify the system (see Sect. 4.1.4 for the theoretical reasoning).

The theoretical material presented in this chapter up to this point is only a review. The following sections are contributions of the author of the thesis.

#### 4.1.4 System identification

In this section, we show that one is able to identify the underlying system given the right number of measurements [50]. Assume that our system has McMillan degree  $n$ , that the number of inputs and outputs is equal ( $m = p$ ) and that the underlying  $\mathbf{D}$  matrix is full rank. Thus, we are looking for a realization of the system of order  $n$  which is minimal (both controllable and observable).

Let us begin our argument by noting that, even though our realizations have  $\mathbf{D} = \mathbf{0}$ , we can also identify systems with a non-zero  $\mathbf{D}$ , because the  $\mathbf{D}$ -term can be incorporated either in the  $\mathbf{B}$  or the  $\mathbf{C}$  matrices. Assume a realization of the system as  $[\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$  with  $\mathbf{E}, \mathbf{A}$  invertible. The following realization

$$\tilde{\mathbf{E}} = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ -\mathbf{I} \end{bmatrix}, \quad \tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \mathbf{D} \end{bmatrix}, \quad \tilde{\mathbf{D}} = \mathbf{0} \quad (4.36)$$

corresponds to a system of order  $n + p$  (i.e., both  $\mathbf{A}$  and  $\mathbf{E}$  are of dimension  $(n + p) \times (n + p)$ ), but a zero  $\mathbf{D}$ -term. One can easily check that the transfer function of the system in the new realization is as before.

Next, we show that if  $n + p$  left tangential data and  $n + p$  right tangential data are provided, and the conditions in Theorem 4.1.1 are satisfied, we recover the system.

The number of free parameters for a rational matrix function  $\mathbf{H}(s) \in \mathbb{C}^{p \times p}$  with  $\mathbf{D} \neq \mathbf{0}$  and McMillan degree  $n$  is  $2np + 2p^2$ , which is obtained by considering the control canonical form [57] of the system. Thus, we can freely choose  $n$  entries on  $p$  rows of  $\mathbf{A}$ ,  $p$  entries on  $p$  rows of  $\mathbf{B}$ ,  $np$  entries of  $\mathbf{C}$ , as well as the  $p^2$  entries of  $\mathbf{D}$ . On the other hand, the number of free parameters in tangential interpolation are  $2(n + p)$  for  $\lambda_i, \mu_j, i, j = 1, \dots, n + p$ , as well as the tangential directions  $\mathbf{r}_i \in \mathbb{C}^{p \times 1}, i =$

$1, \dots, n+p$  and  $\ell_j \in \mathbb{C}^{1 \times p}$ ,  $j = 1, \dots, n+p$ . Note that once the sampling points  $\lambda_i$  and  $\mu_j$  and the sampling directions  $\mathbf{r}_i$  and  $\ell_j$  were chosen, the left and right data  $\mathbf{w}_i$  and  $\mathbf{v}_j$  are set, as they are obtained as  $\mathbf{H}(\lambda_i)\mathbf{r}_i = \mathbf{w}_i$  and  $\ell_j\mathbf{H}(\mu_j) = \mathbf{v}_j$ , respectively. This gives a total number of free parameters of  $2(n+p) + p(n+p) + (n+p)p = 2n + 2p + 2np + 2p^2$ . Thus, the number of the free parameters needed to uniquely define the transfer function matrix is smaller than the number of unknown parameters when  $n+p$  tangential measurements are chosen on the left and on the right.

In the following, we show how the Loewner matrix can be factored in terms of tangential controllability and observability matrices. A similar expression exists for the shifted Loewner matrix. Given  $n+p$  left interpolation data  $\mathbf{v}_j = \ell_j\mathbf{H}(\mu_j)$ ,  $j = 1, \dots, n+p$  and the same number of right interpolation data  $\mathbf{w}_i = \mathbf{H}(\lambda_i)\mathbf{r}_i$ ,  $i = 1, \dots, n+p$ , we obtain

$$\begin{aligned} \mathbb{L}_{j,i} &= \frac{\mathbf{v}_j\mathbf{r}_i - \ell_j\mathbf{w}_i}{\mu_j - \lambda_i} = \frac{\ell_j\mathbf{H}(\mu_j)\mathbf{r}_i - \ell_j\mathbf{H}(\lambda_i)\mathbf{r}_i}{\mu_j - \lambda_i} \\ &= \ell_j \frac{\mathbf{C}(\mu_j\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} - \mathbf{C}(\lambda_i\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} - \mathbf{D}}{\mu_j - \lambda_i} \mathbf{r}_i \\ &= \ell_j \mathbf{C} \frac{(\mu_j\mathbf{E} - \mathbf{A})^{-1} - (\lambda_i\mathbf{E} - \mathbf{A})^{-1}}{\mu_j - \lambda_i} \mathbf{B} \mathbf{r}_i = -\ell_j \mathbf{C}(\mu_j\mathbf{E} - \mathbf{A})^{-1} \mathbf{E}(\lambda_i\mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_i. \end{aligned}$$

Similarly,

$$\begin{aligned} \sigma\mathbb{L}_{j,i} &= \frac{\mu_j\mathbf{v}_i\mathbf{r}_i - \lambda_i\ell_j\mathbf{w}_i}{\mu_j - \lambda_i} = \ell_j \mathbf{C} \frac{\mu_j(\mu_j\mathbf{E} - \mathbf{A})^{-1} - \lambda_i(\lambda_i\mathbf{E} - \mathbf{A})^{-1}}{\mu_j - \lambda_i} \mathbf{B} \mathbf{r}_i + \ell_j \mathbf{D} \mathbf{r}_i \\ &= \ell_j \frac{\mu_j\mathbf{C}(\mu_j\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mu_j\mathbf{D} - \lambda_i\mathbf{C}(\lambda_i\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} - \lambda_i\mathbf{D}}{\mu_j - \lambda_i} \mathbf{r}_i \\ &= -\ell_j \mathbf{C}(\mu_j\mathbf{E} - \mathbf{A})^{-1} \mathbf{A}(\lambda_i\mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_i + \ell_j \mathbf{D} \mathbf{r}_i, \quad j, i = 1, \dots, n+p. \end{aligned}$$

Thus

$$\mathbb{L} = - \underbrace{\begin{bmatrix} \ell_1 \mathbf{C} (\mu_1 \mathbf{E} - \mathbf{A})^{-1} \\ \vdots \\ \ell_{n+p} \mathbf{C} (\mu_{n+p} \mathbf{E} - \mathbf{A})^{-1} \end{bmatrix}}_{\mathbf{O}} \mathbf{E} \underbrace{\begin{bmatrix} (\lambda_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_1 & \dots & (\lambda_{n+p} \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_{n+p} \end{bmatrix}}_{\mathcal{R}}, \quad (4.37)$$

$$\sigma \mathbb{L} = - \underbrace{\begin{bmatrix} \ell_1 \mathbf{C} (\mu_1 \mathbf{E} - \mathbf{A})^{-1} \\ \vdots \\ \ell_{n+p} \mathbf{C} (\mu_{n+p} \mathbf{E} - \mathbf{A})^{-1} \end{bmatrix}}_{\mathbf{O}} \mathbf{A} \underbrace{\begin{bmatrix} (\lambda_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_1 & \dots & (\lambda_{n+p} \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_{n+p} \end{bmatrix}}_{\mathcal{R}} \quad (4.38)$$

+ LDR.

The pairs  $(\mathbf{E}, \mathbf{A}, \mathbf{B})$  and  $(\mathbf{C}, \mathbf{E}, \mathbf{A})$  are controllable and observable, respectively, so provided that the sampling directions are chosen appropriately, the rank of the Loewner matrix is precisely the rank of the underlying  $\mathbf{E}$  matrix, while the rank of the shifted Loewner matrix is  $p$  more than the rank of the underlying  $\mathbf{A}$  matrix, which is as expected from Eq.(4.36). Next we provide an outline on how to choose the sampling tangential directions.

- If  $\mathbf{D}=0$ , then  $\mathbb{L} = -\mathbf{O}\mathbf{E}\mathcal{R}$  and  $\sigma\mathbb{L} = -\mathbf{O}\mathbf{A}\mathcal{R}$  and we can choose the same sampling direction for all sampling points:  $\ell_1 = \ell_2 = \dots = \ell_{n+p}$  and  $\mathbf{r}_1 = \mathbf{r}_2 = \dots = \mathbf{r}_{n+p}$ . This is because  $(\mathbf{E}, \mathbf{A}, \mathbf{B})$  is controllable, so by using a linear combination of the columns of  $\mathbf{B}$ , one can still control the system. Similarly, as  $(\mathbf{C}, \mathbf{E}, \mathbf{A})$  is observable, choosing a linear combination of the rows of  $\mathbf{C}$  still makes the system observable. Denoting  $\ell_1 \mathbf{C}$  by  $\mathbf{c}$  and  $\mathbf{B} \mathbf{r}_1$  by  $\mathbf{b}$ , we have that

$$\mathbb{L} = - \underbrace{\begin{bmatrix} \mathbf{c} (\mu_1 \mathbf{E} - \mathbf{A})^{-1} \\ \vdots \\ \mathbf{c} (\mu_{n+p} \mathbf{E} - \mathbf{A})^{-1} \end{bmatrix}}_{\mathcal{O}} \mathbf{E} \underbrace{\begin{bmatrix} (\lambda_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{b} & \dots & (\lambda_{n+p} \mathbf{E} - \mathbf{A})^{-1} \mathbf{b} \end{bmatrix}}_{\mathcal{R}} \quad (4.39)$$

and similarly for  $\sigma\mathbb{L}$ .

- The statement above does not hold when  $(\mathbf{A}, \mathbf{E})$  has at least one multiple eigenvalue with the same algebraic and geometric multiplicity (eg.,  $\mathbf{E} = \mathbf{A} = \mathbf{I}$ ), as one would need more than one vector  $\mathbf{b}$  to make  $(\mathbf{E}, \mathbf{A}, \mathbf{b})$  controllable. Similarly, more than one  $\mathbf{c}$  is required to make  $(\mathbf{c}, \mathbf{E}, \mathbf{A})$  observable. Hence, the sampling directions for each evaluating point should be different. Note that having Jordan blocks associated with the same eigenvalue is not an issue from the controllability/observability point of view.
- If  $\mathbf{D} \neq \mathbf{0}$ , Eq.(4.38) shows that the  $\mathbf{D}$ -term cannot be recovered unless we choose at least  $p$  linearly independent vectors  $\mathbf{r}_i$  and  $\ell_j$ . In particular, one can choose the sampling directions as unit vectors of dimension  $p$ .

Theorem 4.1.1 provides a realization of the system in terms of the Loewner matrix pencil and the  $\mathbf{V}$  and  $\mathbf{W}$  matrices. Therefore, this method constructs a descriptor-form representation of an underlying dynamical system exclusively from the data, just by arranging it in a convenient form, with no further manipulations involved.

**Example 4.1.4.** Consider the following transfer function with  $\mathbf{D} \neq \mathbf{0}$ [50]

$$\mathbf{H}(s) = \frac{1}{s^2 + 5s + 6} \begin{bmatrix} s & -6 \\ 1 & s + 5 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}. \quad (4.40)$$

This purpose of this example is to show how the  $\mathbf{D}$  term affects the realization. We are given matrix data at  $i$ ,  $2i$ ,  $3i$ ,  $4i$  and their complex conjugates (i.e.,  $\mathbf{H}(i), \dots, \mathbf{H}(4i)$  and  $\mathbf{H}(-i) = \overline{\mathbf{H}(i)}, \dots, \mathbf{H}(-4i) = \overline{\mathbf{H}(4i)}$ ). We construct tangential data as

$$k=4, (\lambda, \mathbf{r}, \mathbf{w}) = \{(i, \mathbf{e}_1, \mathbf{H}(i)\mathbf{e}_1), (-i, \mathbf{e}_1, \mathbf{H}(-i)\mathbf{e}_1), (3i, \mathbf{e}_2, \mathbf{H}(3i)\mathbf{e}_2), (-3i, \mathbf{e}_2, \mathbf{H}(-3i)\mathbf{e}_2)\}$$

$$h=4, (\mu, \ell, \mathbf{v}) = \{(2i, \mathbf{e}_1^T, \mathbf{e}_1^T \mathbf{H}(2i)), (-2i, \mathbf{e}_1^T, \mathbf{e}_1^T \mathbf{H}(-2i)), \\ (4i, \mathbf{e}_2^T, \mathbf{e}_2^T \mathbf{H}(4i)), (-4i, \mathbf{e}_2^T, \mathbf{e}_2^T \mathbf{H}(-4i))\}$$

and obtain the following realization, after performing the appropriate basis change:

$$\mathbf{E}_r = -\mathbb{L} = - \begin{bmatrix} -\frac{1}{130} & -\frac{2}{65} & 0 & -\frac{3}{26} \\ \frac{4}{65} & -\frac{7}{130} & \frac{1}{13} & -\frac{5}{26} \\ -\frac{1}{250} & \frac{1}{125} & -\frac{2}{325} & \frac{19}{650} \\ -\frac{4}{125} & \frac{3}{125} & -\frac{38}{975} & \frac{83}{975} \end{bmatrix}, \mathbf{A}_r = -\sigma\mathbb{L} = - \begin{bmatrix} \frac{159}{130} & -\frac{1}{130} & \frac{29}{13} & 0 \\ \frac{1}{65} & \frac{4}{65} & 0 & \frac{3}{13} \\ \frac{743}{250} & -\frac{1}{250} & \frac{1291}{325} & -\frac{6}{325} \\ \frac{2}{125} & -\frac{4}{125} & \frac{8}{325} & -\frac{38}{325} \end{bmatrix}, \\ \mathbf{B}_r = \mathbf{V} = \begin{bmatrix} \frac{31}{26} & \frac{49}{26} \\ -\frac{1}{26} & -\frac{15}{26} \\ \frac{149}{50} & \frac{203}{50} \\ \frac{1}{25} & \frac{7}{25} \end{bmatrix}, \mathbf{C}_r = \mathbf{W} = \begin{bmatrix} \frac{11}{10} & \frac{1}{10} & \frac{27}{13} & \frac{5}{13} \\ \frac{31}{10} & -\frac{1}{10} & \frac{161}{39} & -\frac{14}{39} \end{bmatrix}, \mathbf{D}_r = \mathbf{0}_2.$$

The singular values of  $\mathbf{E}_r$ , which is the Loewner matrix up to a sign change, are  $\frac{487}{1811}$ ,  $\frac{345}{6374}$ , 0, 0, while those of  $\mathbf{A}_r$  are  $\frac{2349}{422}$ ,  $\frac{397}{1246}$ ,  $\frac{364}{1359}$ ,  $\frac{44}{55193}$ , so indeed our realization has order  $4 = 2 + 2 = n + p$  with  $\mathbf{A}_r$  full rank and  $\mathbf{E}_r$  of rank  $2 = n$ . Moreover, the eigenvalues of the matrix pencil are  $-5 \cdot 10^{14}$ ,  $-2 \cdot 10^{14}$ ,  $-3$ ,  $-2$ . The first two correspond to poles at infinity, while the last two are precisely the poles of our original system (given by the roots of the equation  $s^2 + 5s + 6 = 0$ ).

## 4.2 Implementation

After demonstrating the application of the theoretical concepts described in the previous sections to the problem of modeling measured multi-port frequency response measurements, this section presents several efficient implementation approaches [43, 50]. These implementations are contributions of the author of this thesis. Data sets contain  $N$  samples of the measured frequency response,  $\mathbf{H}^{(i)}$ , at frequency samples  $j\omega_i$ ,

for  $i = 1, \dots, N$ . To obtain a real system, the condition  $\overline{\mathbf{H}(s)} = \mathbf{H}(\bar{s})$  needs to be satisfied, so the response at the complex conjugate values of the sample points  $-j\omega_i$  should equal the complex conjugates of the measurement at  $j\omega_i$ , namely  $\overline{\mathbf{H}^{(i)}}$ .

### 4.2.1 Complex approach

Setting the tangential directions  $\mathbf{r}_i$  and  $\ell_i$  as rows and columns of the identity matrix of dimension  $p$  leads to the right and left data  $\mathbf{w}_i$  and  $\mathbf{v}_j$  being rows and columns of the measured frequency response matrices, respectively [43].

The right interpolation data can be chosen as

$$(\lambda_i = j\omega_i, \mathbf{r}_i, \mathbf{w}_i = \mathbf{H}^{(i)}\mathbf{r}_i), \quad (4.41)$$

where  $\omega_i = 2\pi f_i \in \mathbb{R}$ . The direction  $\mathbf{r}_i$  is  $\mathbf{r}_i = \mathbf{e}_m \in \mathbb{R}^{p \times 1}$ , with  $m = p$  for  $i = p \cdot c_1$  and  $m = 1, \dots, p-1$  for  $i = p \cdot c_1 + m$ , for some  $c_1 \in \mathbb{Z}$ , where  $\mathbf{e}_m$  denotes the  $m$ -th column of the identity matrix  $\mathbf{I}_p$ . Consequently, the right data are  $\mathbf{w}_i = \mathbf{H}^{(i)}\mathbf{r}_i = \mathbf{H}_{:,m}^{(i)} \in \mathbb{C}^{p \times 1}$  (i.e., the  $m$ -th column of  $\mathbf{H}^{(i)}$ ) for  $i = 1, \dots, N$ .

The left interpolation data are constructed as

$$(\mu_i = -j\omega_i, \ell_i, \mathbf{v}_i = \ell_i \overline{\mathbf{H}^{(i)}}), \quad (4.42)$$

where  $\ell_i = \mathbf{r}_i^T \in \mathbb{R}^{1 \times p}$ . When  $\ell_i$  is a unit vector  $\mathbf{e}_m^T$ , the left data  $\mathbf{v}_i$  is  $\mathbf{v}_i = \ell_i \overline{\mathbf{H}^{(i)}} = \overline{\mathbf{H}_{m,:}^{(i)}} \in \mathbb{C}^{1 \times p}$  (i.e., the  $m$ -th row of the complex conjugate value of  $\mathbf{H}^{(i)}$ , where  $m$  is  $p$  for  $i = p \cdot c_1$  and  $1, \dots, p-1$  for  $i = p \cdot c_1 + m$ , for some  $c_1 \in \mathbb{Z}$ ) for  $i = 1, \dots, N$ .

After the tangential data have been identified, the Loewner and shifted Loewner matrices are built as in Eq.(4.21) and (4.23).

### 4.2.2 Real approach

Even though the complex approach uses the information at the complex conjugate values of the points, the realization matrices are complex and the poles, which, ac-

According to Theorem 4.1.1, are the eigenvalues of the Loewner pencil, may not come in complex conjugate pairs due to numerical issues [43]. To guarantee a resulting system which is real, the right interpolation data can be chosen as

$$\left( j\omega_i, -j\omega_i; \mathbf{r}_i, \mathbf{r}_i; \mathbf{w}_i = \mathbf{H}^{(i)} \mathbf{r}_i, \bar{\mathbf{w}}_i = \overline{\mathbf{H}^{(i)} \mathbf{r}_i} \right), \quad (4.43)$$

where  $\omega_i = 2\pi f_i \in \mathbb{R}$ . The right directions may be chosen as  $\mathbf{r}_i = \mathbf{e}_m \in \mathbb{R}^{p \times 1}$ , with  $m = p$  for  $i = p \cdot c_1$  and  $m = 1, \dots, p-1$  for  $i = p \cdot c_1 + m$ , for some  $c_1 \in \mathbb{Z}$ , where  $\mathbf{e}_m$  denotes the  $m$ -th column of the identity matrix  $\mathbf{I}_p$ , for  $i = 1, \dots, \frac{N}{2}$ . The left interpolation data is constructed as

$$\left( j\omega_{i+\frac{N}{2}}, -j\omega_{i+\frac{N}{2}}; \ell_i, \ell_i; \mathbf{v}_i = \ell_i \mathbf{H}^{(i+\frac{N}{2})}, \bar{\mathbf{v}}_i = \overline{\ell_i \mathbf{H}^{(i+\frac{N}{2})}} \right). \quad (4.44)$$

The left directions  $\ell_i$  may be chosen as before. Without loss of generality, we assumed an even number of samples. After tangential data are constructed, the Loewner and shifted Loewner matrices are built using Eq.(4.21) and (4.23). Next, a change of basis needs to be performed to ensure that the matrix entries are real:  $\hat{\Lambda} = \hat{\Pi}^* \Lambda \hat{\Pi}$ ,  $\hat{M} = \hat{\Pi}^* M \hat{\Pi}$ ,  $\hat{L} = \hat{\Pi}^* L$ ,  $\hat{V} = \hat{\Pi}^* V$ ,  $\hat{R} = R \hat{\Pi}$ ,  $\hat{W} = W \hat{\Pi}$ ,  $\hat{L} = \hat{\Pi}^* L \hat{\Pi}$ ,  $\hat{\sigma} L = \hat{\Pi}^* \sigma L \hat{\Pi}$ , where

$$\hat{\Pi} = \text{blkdiag} [\Pi, \dots, \Pi] \in \mathbb{C}^{N \times N}, \quad \Pi = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}.$$

Thus, each  $2 \times 2$  block can be written explicitly as:

$$\begin{aligned} \hat{\Lambda}_i &= \begin{bmatrix} 0 & \omega_i \\ -\omega_i & 0 \end{bmatrix}, \quad \hat{M}_i = \begin{bmatrix} 0 & \omega_{i+\frac{N}{2}} \\ -\omega_{i+\frac{N}{2}} & 0 \end{bmatrix}, \\ \hat{L}_i &= \sqrt{2} \begin{bmatrix} \ell_i \\ 0 \end{bmatrix}, \quad \hat{R}_i = \sqrt{2} \begin{bmatrix} \mathbf{r}_i & 0 \end{bmatrix}, \\ \hat{V}_i &= \sqrt{2} \begin{bmatrix} \Re(\mathbf{v}_i) \\ -\Im(\mathbf{v}_i) \end{bmatrix}, \quad \hat{W}_i = \sqrt{2} \begin{bmatrix} \Re(\mathbf{w}_i) & \Im(\mathbf{w}_i) \end{bmatrix}. \end{aligned}$$



To build  $\hat{\mathbf{L}}$  and  $\hat{\sigma}\hat{\mathbf{L}}$  efficiently in Matlab, without the use of “for” loops, we wish to transform the Sylvester equations

$$\hat{\mathbf{L}}\hat{\mathbf{A}} - \hat{M}\hat{\mathbf{L}} = \hat{\mathbf{L}}\hat{\mathbf{W}} - \hat{\mathbf{V}}\hat{\mathbf{R}}, \quad (4.45)$$

$$\hat{\sigma}\hat{\mathbf{L}}\hat{\mathbf{A}} - \hat{M}\hat{\sigma}\hat{\mathbf{L}} = \hat{\mathbf{L}}\hat{\mathbf{W}}\hat{\mathbf{A}} - \hat{M}\hat{\mathbf{V}}\hat{\mathbf{R}} \quad (4.46)$$

to equations of the form  $\hat{\mathbf{L}}\mathbf{X} - \mathbf{Y}\hat{\mathbf{L}} = \mathbf{Z}$  where  $\mathbf{X}$  and  $\mathbf{Y}$  are diagonal. Note that  $\hat{\mathbf{A}}^2$  and  $\hat{M}^2$  are diagonal, so by multiplying Eq.(4.45) on the left by  $\hat{M}$  and on the right by  $\hat{\mathbf{A}}$ , and adding the resulting equations gives the desired expression:

$$\hat{\mathbf{L}}\hat{\mathbf{A}}^2 - \hat{M}^2\hat{\mathbf{L}} = \hat{M}(\hat{\mathbf{L}}\hat{\mathbf{W}} - \hat{\mathbf{V}}\hat{\mathbf{R}}) + (\hat{\mathbf{L}}\hat{\mathbf{W}} - \hat{\mathbf{V}}\hat{\mathbf{R}})\hat{\mathbf{A}}.$$

Similarly, the shifted Loewner matrix with real entries is the solution of the following Sylvester equation:

$$\hat{\sigma}\hat{\mathbf{L}}\hat{\mathbf{A}}^2 - \hat{M}^2\hat{\sigma}\hat{\mathbf{L}} = \hat{M}(\hat{\mathbf{L}}\hat{\mathbf{W}} - \hat{\mathbf{V}}\hat{\mathbf{R}})\hat{\mathbf{A}} + \hat{\mathbf{L}}\hat{\mathbf{W}}\hat{\mathbf{A}}^2 - \hat{M}^2\hat{\mathbf{V}}\hat{\mathbf{R}}.$$

### 4.2.3 Real alternative approach

We present an alternative ordering in which the tangential data in the real approach are arranged, which yields better results in the entire frequency band. Recall that interpolation is exact at  $\lambda_i$ , since these are the expansion points in the Lagrange basis, so it is better to choose them distributed over the entire frequency range, rather than only at the beginning. The right interpolation data can be chosen as

$$\left(\lambda_i = j\omega_i, \lambda_{i+1} = -j\omega_i; \mathbf{r}_i, \mathbf{r}_{i+1} = \mathbf{r}_i; \mathbf{w}_i = \mathbf{H}^{(i)}\mathbf{r}_i, \mathbf{w}_{i+1} = \bar{\mathbf{w}}_i = \bar{\mathbf{H}}^{(i)}\mathbf{r}_i\right), i=1, 3, \dots, N \quad (4.47)$$

where  $\omega_i = 2\pi f_i \in \mathbb{R}$ . The right sampling directions may be chosen as  $\mathbf{r}_1 = \mathbf{r}_2 = \mathbf{e}_1 \in \mathbb{R}^{p \times 1}$ ,  $\mathbf{r}_3 = \mathbf{r}_4 = \mathbf{e}_2, \dots, \mathbf{r}_{2p-1} = \mathbf{r}_{2p} = \mathbf{e}_p, \mathbf{r}_{2p+1} = \mathbf{r}_{2p+2} = \mathbf{e}_1, \mathbf{r}_{2p+3} = \mathbf{r}_{2p+4} = \mathbf{e}_2, \dots$ , where  $\mathbf{e}_m$  denotes the  $m$ -th column of the identity matrix  $\mathbf{I}_p$ ,  $m = 1, \dots, p$ . Consequently, the right data are columns of  $\mathbf{H}^{(i)}$  and  $\bar{\mathbf{H}}^{(i)}$ , eg.  $\mathbf{w}_1 = \mathbf{H}_{:,1}^{(1)} \in \mathbb{C}^{p \times 1}$ ,  $\mathbf{w}_2 = \bar{\mathbf{H}}_{:,1}^{(1)}$ ,  $\mathbf{w}_3 = \mathbf{H}_{:,2}^{(2)}$ ,  $\mathbf{w}_4 = \bar{\mathbf{H}}_{:,2}^{(2)}$ , etc. The left interpolation data is constructed as

$$\left(\mu_i=j\omega_{i+1}, \mu_{i+1}=-j\omega_{i+1}; \ell_i, \ell_{i+1}=\ell_i; \mathbf{v}_i=\ell_i\mathbf{H}^{(i+1)}, \mathbf{v}_{i+1}=\bar{\mathbf{v}}_i=\ell_i\bar{\mathbf{H}}^{(i+1)}\right), i=1, 3, \dots, N \quad (4.48)$$

The left sampling directions  $\ell_i$  may be chosen as rows of the identity matrix  $\mathbf{I}_p$ , so the left data are rows of the admittance parameters  $\mathbf{H}^{(i)}$  and  $\bar{\mathbf{H}}^{(i)}$ .

Without loss of generality, we assumed that we have an even number of samples. After tangential data are constructed, the Loewner and shifted Loewner matrices are built according to Eq.(4.21) and (4.23), and the same change of basis as in Sect. 4.2.2 can be performed to obtain matrices with real entries.

#### 4.2.4 SVD implementation

The first idea (as presented in [44]) is to use all measurements to construct the Loewner matrix pencil, in the complex approach, presented in Section 4.2.1, or in one of the real approaches, presented in Sect. 4.2.2 and 4.2.3. Theorem 4.1.1 ensures recovery of the system when the Loewner matrix pencil is regular and the measurements are noise-free. However, when too many measurements are available, the pencil is singular, so one needs to eliminate the singular part via a rank revealing factorization. Under the assumption that  $\forall x \in \{\lambda_i\} \cup \{\mu_i\}$

$$\text{rank}(x\mathbb{L} - \sigma\mathbb{L}) =: r, \quad (4.49)$$

one can perform the singular value decomposition:

$$x\mathbb{L} - \sigma\mathbb{L} = \mathbf{Y}_1\Sigma\mathbf{X}_1^*, \quad (4.50)$$

where  $\text{rank}(x\mathbb{L} - \sigma\mathbb{L}) = \text{rank}(\Sigma) =: r$ ,  $\mathbf{Y}_1, \mathbf{X}_1 \in \mathbb{C}^{k \times r}$ , with  $r$ , the *dimension of the regular part* of  $x\mathbb{L} - \sigma\mathbb{L}$ . For strictly proper systems,  $r$  is precisely  $n$ , the order of the underlying system, while for proper systems,  $r = n + \text{rank}(\mathbf{D})$ . Using the singular vectors as projectors, the realization [44] is given as  $\mathbf{E} = -\mathbf{Y}_1^*\mathbb{L}\mathbf{X}_1$ ,  $\mathbf{A} = -\mathbf{Y}_1^*\sigma\mathbb{L}\mathbf{X}_1$ ,  $\mathbf{B} = \mathbf{Y}_1^*\mathbf{V}$ ,  $\mathbf{C} = \mathbf{W}\mathbf{X}_1$ ,  $\mathbf{D} = \mathbf{0}$ . The choice of  $x$  is not an issue since the pencil  $(\sigma\mathbb{L}, \mathbb{L})$  loses rank only when  $x$  is one of its eigenvalues. However, Theorem 4.1.1 includes

this assumption ( $\mu_j, \lambda_i \notin \lambda(\sigma\mathbb{L}, \mathbb{L})$ ), which holds in the case of modeling frequency domain data as  $x$  is on the imaginary axis, while the underlying system is stable (so the eigenvalues of  $(\sigma\mathbb{L}, \mathbb{L})$  are in the left half plane).

Nevertheless, real-world measurements are noisy, so the zero singular values of the singular pencil are corrupted by noise. As pointed out in [58], the singular values corresponding to zero singular values in the unperturbed pencil are larger than the noise by a factor proportional to the square root of the number of samples. Consequently, one can perform the SVD of the matrix  $x\mathbb{L} - \sigma\mathbb{L}$  for some  $x \in \{\lambda_i\} \cup \{\mu_i\}$  and identify the order of the underlying system based on a large drop in the singular values, or the above criterion. The pseudocode is included in Appendix B.1.

To extract the regular part of the pencil (for reasonable matrix dimensions), one may also use GUPTRI [59], [60], which computes the generalized upper triangular form of a matrix pencil  $\mathbf{A} - \lambda\mathbf{B}$  [46].

The SVD approach is global, since it uses all samples to identify the model, but the computational cost for data sets with a large number of samples  $N$  is big, as the cost of the SVD of the matrix  $x\mathbb{L} - \sigma\mathbb{L}$  scales with  $N^3$ . The adaptive approach presented next is not global, as it only uses some samples to build the system, but it is faster for large  $N$  and  $p$ .

### 4.2.5 Adaptive implementation

In the next proposed implementation, we choose samples from the available ones adaptively to construct the desired model [47, 43]. A common technique would be to assess the error between the current model and the data from the magnitude of all entries of the  $p \times p$  scattering matrix. Nevertheless, because this technique is expensive when the number of ports  $p$  is large, we use the singular values of the error matrices defined in Eq.(3.7) instead. They encode the errors in all entries and, moreover, have the advantage that they are real and positive. Recall that the 2-norm

of a matrix is given by the largest singular value, so using the first singular value of the error matrices as an error criterion is justified. If the first singular value is small for all samples, the overall errors are small and the model is accurate.

By using block processing and adding  $p$  samples at each step, our approach is better suited for large  $p$ , as the computational complexity scales with  $O(Np^3)$ .

The pseudocode is shown in Appendix B.2. Algorithm 2 uses the original measurements as right data and their complex conjugate values as left data, so the resulting model may have complex entries. It can be extended to the real approach (see Appendix 4.2.2) by using half of the measurements added at each step, together with their complex conjugate values, as right data and the rest as left data.

#### 4.2.6 Stability and passivity

Our algorithms identify the underlying system; therefore, for data sets describing real-world systems, the models are stable (after extracting the necessary  $\mathbf{D}$ -term). If the underlying system is symmetric, by choosing the left tangential directions as transpose of the right tangential directions (see Sect. 4.2.1), passivity is preserved. In general, passivity is not enforced by construction, but for slight passivity violations (of less than 0.1%), the easiest way to obtain a passive model is to divide either the  $\mathbf{B}$  or the  $\mathbf{C}$  state-space matrix of the corresponding realization by this maximum. Out of band violations can be corrected by an a posteriori passivation enforcement, for example [61]. Passivity can also be enforced in the current framework, as in [6].

The concepts presented here are more general than [5], which employs Nevanlinna-Pick interpolation for the bounded-real interpolation of S-parameters. In particular, as shown in [6], passivity is enforced by interpolating at the original points  $(s_i, \mathbf{H}(s_i))$ , together with the associated *mirror-image* points  $(-s_i^*, \frac{1}{\mathbf{H}(-s_i^*)})$ . Hence this method cannot recover the original system unless the latter satisfies the mirror-image interpolation constraints, which is generally not the case [43].

### 4.2.7 Computational Complexity

Recall the following quantities:  $N$ , the number of samples,  $p$ , the number of ports,  $k$ , the number of starting poles for vector fitting (VF) and  $n$ , the dimension of the resulting model (the vector fitting algorithm is explained in Chapter 6). For easily comparing the complexities, we assume that the dimension of the macromodel  $n$  is as in column-wise VF, namely  $k \cdot p$ . This assumption is not necessary in our SVD and adaptive approaches. This section gives expressions for the computational complexity of each algorithm in terms of these quantities.

#### 4.2.7.1 Vector fitting

The complexity of the column-wise implementation of the vector fitting algorithm is

$$\mathcal{O}(N_2 N p^4 k^2) = \mathcal{O}\left(N_2 N n^4 \frac{1}{k^2}\right),$$

where  $N_2$  is the number of iterations used to fit each column. This expression is due to the fact that a least-squares problem with a matrix of dimensions  $(Np) \times (k(p+1))$  is solved when fitting each of the  $p$  columns using  $N_2$  iterations.

#### 4.2.7.2 SVD implementation approach

The complexity is

$$\mathcal{O}(N^3 + N^2 pk + N p^2 k^2) = \mathcal{O}(N^3 + N^2 k + N n^2).$$

The first term is from computing the SVD of  $x\mathbb{L} - \sigma\mathbb{L}$ , which is a matrix of size  $N$ , while the next terms come from computing the projected reduced systems.

#### 4.2.7.3 Adaptive implementation approach

The most expensive operation is the evaluation of the transfer function of the current model, for which a matrix inversion is necessary, for all samples, so the complexity is

$$\mathcal{O}(Np^3k^4) = \mathcal{O}\left(N\frac{n^4}{k}\right).$$

#### 4.2.7.4 Discussion of the algorithms' computational complexity

As expected, our approaches are better suited for data sets obtained from devices with a large number of ports, as the complexity depends on  $p^2$  or  $p^3$ , as opposed to column-wise VF, for which the computational cost scales with  $p^4$ . As for the number of samples available in the data set, all algorithms scale linearly with  $N$ , except for the SVD approach, which depends on the third power, making it suitable for sets with fewer samples, but with a large number of ports.

### 4.3 Noisy measurements

As mentioned previously, system identification, as described in Sect. 4.1.4, is only possible if measurements are noise free. However, in real applications, measurement devices are only able to output a certain number of significant digits of the measured quantities. Therefore, measurements are corrupted with noise. Next, we investigate the effects of noise on the SVD implementation of the Loewner framework.

**Problem Statement:** An LTI system  $\Sigma$  models a set with  $N$  *noisy* samples of a device with  $p$  ports ( $f_i, \tilde{\mathbf{H}}^{(i)} = \mathbf{H}^{(i)} + \mathbf{N}^{(i)}$ ),  $i = 1, \dots, N$ , where  $f_i \in \mathbb{R}$  (the frequency where we measure),  $\mathbf{H}^{(i)}, \mathbf{N}^{(i)} \in \mathbb{C}^{p \times p}$  (the true measurement and the noise matrix, respectively), if the value of the associated transfer function evaluated at  $j \cdot 2\pi f_i$  is close to the noisy measurement:  $\mathbf{H}(j \cdot 2\pi f_i) \approx \tilde{\mathbf{H}}^{(i)}$ ,  $i = 1, \dots, N$  [49].

#### 4.3.1 The Loewner matrix pencil under noisy data

Noisy frequency domain data is contained in the  $\mathbf{V}$  and  $\mathbf{W}$  data matrices, while  $\Lambda$ ,  $M$ ,  $\mathbf{R}$  and  $\mathbf{L}$  are assumed not affected by noise. We denote the matrices  $\mathbf{V}$  and  $\mathbf{W}$  containing noisy measurements by  $\tilde{\mathbf{V}}$  and  $\tilde{\mathbf{W}}$ . Thus, the Loewner matrix is affected

by noise, but still satisfies a Sylvester equation:  $\tilde{\mathbf{L}}\Lambda - M\tilde{\mathbf{L}} = \mathbf{L}\tilde{\mathbf{W}} - \tilde{\mathbf{V}}\mathbf{R}$ , so

$$\underbrace{(\tilde{\mathbf{L}} - \mathbf{L})}_{\Delta\mathbf{L}} \Lambda - M \underbrace{(\tilde{\mathbf{L}} - \mathbf{L})}_{\Delta\mathbf{L}} = \mathbf{L} \underbrace{(\tilde{\mathbf{W}} - \mathbf{W})}_{\Delta\mathbf{W}} - \underbrace{(\tilde{\mathbf{V}} - \mathbf{V})}_{\Delta\mathbf{V}} \mathbf{R}. \quad (4.51)$$

Similarly, for the shifted Loewner matrix, we have that

$$(\Delta\sigma\mathbf{L})\Lambda - M(\Delta\sigma\mathbf{L}) = \mathbf{L}(\Delta\mathbf{W})\Lambda - M(\Delta\mathbf{V})\mathbf{R},$$

where  $\Delta\mathbf{L}$  and  $\Delta\sigma\mathbf{L}$  are perturbations introduced by noise. Thus, the poles of the system recovered from noisy data, which are the generalized eigenvalues of the matrix pencil  $(\sigma\tilde{\mathbf{L}}, \tilde{\mathbf{L}}) = (\sigma\mathbf{L} + \Delta\sigma\mathbf{L}, \mathbf{L} + \Delta\mathbf{L})$ , are perturbations of the original values, which are the eigenvalues of  $(\sigma\mathbf{L}, \mathbf{L})$ .

### 4.3.2 Proposed Improvements

We investigate the performance of the SVD implementation in the real alternative approach when dealing with noisy measurements in a controlled experiment. Based on its performance, improvements are proposed [49]. In the experiments, we introduce *random noise relative* to the entries in  $\mathbf{H}^{(i)}$ :  $\mathbf{N}^{(i)} = \mathbf{H}^{(i)} .* 10^{-SNR/10}(\text{randn}(p) + j \cdot \text{randn}(p))$ , where  $.*$  is the entry-by-entry multiplication in MATLAB.

The accuracy of the models is assessed using two error measures:

- the normalized  $\mathcal{H}_\infty$ -norm of the error system, which is the largest singular value of all error matrices (defined in Eq.(3.7)) divided by the largest singular value of all measured frequency response matrices:

$$\mathcal{H}_\infty \text{ error} = \frac{\max_{i=1\dots k} \sigma_1(\mathbf{H}(j\omega_i) - \mathbf{H}^{(i)})}{\max_{i=1\dots k} \sigma_1(\mathbf{H}^{(i)})},$$

- the normalized  $\mathcal{H}_2$ -norm of the error system:

$$\mathcal{H}_2 \text{ error} = \sqrt{\frac{\sum_{i=1}^k \|\mathbf{H}(j\omega_i) - \mathbf{H}^{(i)}\|_F^2}{\sum_{i=1}^k \|\mathbf{H}^{(i)}\|_F^2}},$$

where  $\|\cdot\|_F^2$  stands for the square of the Frobenius norm, namely the sum of the magnitude squared of all  $p^2$  entries, which, for the error matrix, is:

$$\|\mathbf{H}(j\omega_i) - \mathbf{H}^{(i)}\|_F^2 = \sum_{k_1=1}^p \sum_{k_2=1}^p \left| \mathbf{H}_{k_1, k_2}(j\omega_i) - \mathbf{H}_{k_1, k_2}^{(i)} \right|^2.$$

The first error measure evaluates the maximum deviation in the singular values, while the second one evaluates the error in the magnitude of all entries, proving to be a good estimate of the overall performance of the model.

For comparison, we compute the  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$ -norm errors of the noise:

- the  $\mathcal{H}_\infty$ -norm of the noise we introduced:  $\mathcal{H}_\infty$  norm =  $\max_{i=1\dots k} \sigma_1(\mathbf{N}^{(i)})$ ,
- the  $\mathcal{H}_2$ -norm of the noise we introduced:  $\mathcal{H}_2$  norm =  $\sqrt{\sum_{i=1}^k \|\mathbf{N}^{(i)}\|_F^2}$ .

We consider an a-priori known system of order 14 with 2 ports. We generate 134 measurements of its transfer function for frequencies between  $10^{-1}$  and  $10^1$  rad/sec.

#### 4.3.2.1 Noise-free case

When we create measurements of the transfer function and do not introduce any noise, we can immediately identify the order of the system from the drop of the singular values of the Loewner and shifted Loewner matrices and recognize that there is no underlying  $\mathbf{D}$ -term (Fig. 4.1(a)). We employed the SVD real alternative approach in these numerical examples. Using the singular vectors associated to the non-zero singular values as projectors, the original system is recovered (Table 4.1).

$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
6.9291e-14	1.6246e-14

Table 4.1: Results for noise-free data

	$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
Model	1.8032e-7	1.5312e-7
Noise	5.5257e-8	2.7242e-7

Table 4.2: Results for  $SNR = 80$



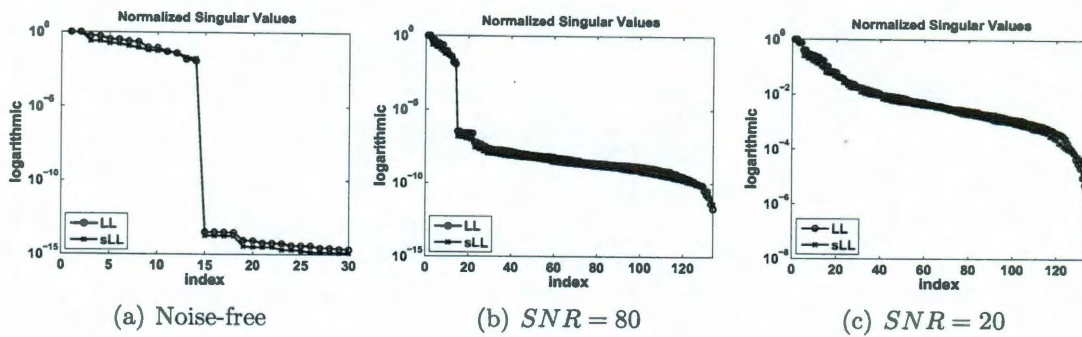


Figure 4.1: Singular value drop of the Loewner matrix pencil

#### 4.3.2.2 $SNR = 80$

For small amounts of noise added to our measurements, we still notice a decay in the singular values of the Loewner matrix pencil (Fig. 4.1(b)), suggesting that the underlying system is of order 14. The singular values which used to be zero are now perturbed by noise. Table 4.2 shows the errors for the order 14 system built using the SVD implementation from Sect. 4.2.4 in the real alternative approach (Sect. 4.2.3) when compared to the noise errors.

#### 4.3.2.3 $SNR = 20$

When the added relative noise is large, the order of the underlying system can no longer be identified from the singular value decay of the Loewner pencil (Fig. 4.1(c)).

Thus, it is instructive to look at *stabilization diagrams* [62, 9]. They compare the location of the poles of the system when incrementally increasing the order of the model. Physical poles are distinguished from spurious ones based on a criterion involving the difference between the poles obtained from two consecutive orders. If the differences are within pre-set limits, the pole is labeled as physical.

In our case, the order is given by the number of singular values retained in the SVD truncation. The stabilization diagram in Fig. 4.2 show the absolute value of the imaginary part of the poles (on the x axis) versus the different model orders (on the y axis). Pluses are stable poles, while stars are unstable poles. Moreover, we

highlighted as black circles the poles which are obtained when truncating at order 14 (which is the true order of the system) and as green circles, the original poles.

For  $SNR = 80$ , the diagram (Fig. 4.2(a)) shows that the poles of the order 14 model approximate well the original ones. Once physical poles were identified, they are present for all subsequent orders higher than that of the underlying system. For  $SNR = 20$ , the stabilization diagram in Fig. 4.2(b) is not as clear as before. Truncation at order 14 has occurred too early, since not all physical poles are present and it is necessary to go to order 30 and higher to obtain relatively good approximations of the original poles. This diagram suggests that it is necessary to construct a high order model from our noisy data. However, such a high-dimensional model would have unstable poles, as well as others that are non-physical, so a reduction step is necessary to eliminate the spurious part of the model.

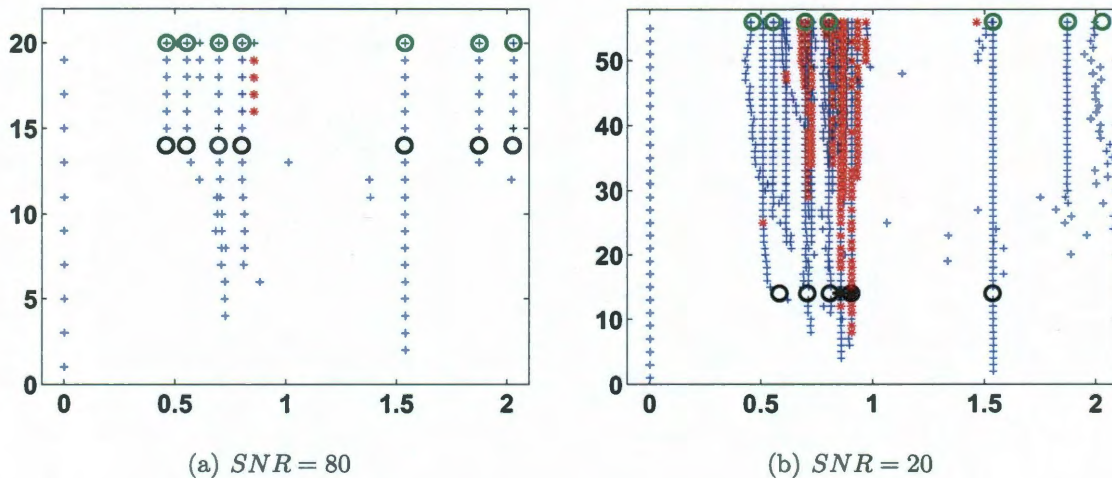


Figure 4.2: Stabilization diagrams

In the civil and mechanical engineering communities, stabilization diagrams are built to reveal the physical poles as those that do not move too much in the complex plane with increasing the order of the model. This is based on the intuition that noise in the data should not affect the underlying system poles. Therefore, stable poles with almost constant imaginary part for different model orders are good candidates

for physical poles. However, for  $SNR = 20$ , some stable poles with almost constant imaginary part are not physical. Thus, this criterion gives, apart from the physical poles, others which are spurious and try to fit the noise. This suggests employing another criterion to detect the true poles of the underlying system.

Below, we investigate two alternatives. The first is to compute the 2-norm of the residue matrices of the high-order model and retain the poles with the largest residue norms. For descriptor systems, residues are computed as:

$$\mathbf{R}_i = (\mathbf{C}\mathbf{x}_i)(\mathbf{y}^*\mathbf{E}\mathbf{x}_i)^{-1}(\mathbf{y}_i^*\mathbf{B}) \quad (4.52)$$

where  $\mathbf{x}_i$  and  $\mathbf{y}_i$  are the right and left eigenvectors associated to the eigenvalue  $\lambda_i$  of the matrix pencil  $(\mathbf{A}, \mathbf{E})$ :  $\mathbf{A}\mathbf{x}_i = \lambda_i\mathbf{E}\mathbf{x}_i$ ,  $\mathbf{y}_i^*\mathbf{A} = \lambda_i\mathbf{y}_i^*\mathbf{E}$ . The second is to preserve the most dominant poles, where dominance is measured by the quantities  $q_i$  defined as:

$$q_i = \frac{\|\mathbf{R}_i\|_2}{\text{Re}(\lambda_i)}. \quad (4.53)$$

Both criteria are motivated by the pole-residue expansion of the transfer function:  $\mathbf{H}(s) = \sum_{i=1}^n \frac{\mathbf{R}_i}{s-\lambda_i}$ , as a pole with large residue norm and/or large dominance quantity contributes more to the response, while the rest do not influence it very much.

For  $SNR = 80$ , it is clear that the first 14 poles are the physical ones from the residue norms, as well as the dominance quantities (Table 4.3).

For  $SNR = 20$ , the first 14 poles sorted according to the residue norms are approximations of the original ones. If sorted according to the dominance criterion, this would not be the case (non-physical poles 19, 20 are more dominant than recovered poles 13, 14). Tables 4.3 and 4.4 also include a column for the minimum distance between each recovered pole and all original ones. All distances are in the range of the noise level. For  $SNR = 80$ , the shortest distances are between those poles with the largest residues (and most dominant) and the original poles. This is not true for  $SNR = 20$ , as the non-physical poles 19, 20 are closer than the recovered poles 3, 4.

i	original poles $\lambda_i$	$\tilde{\lambda}_i$	$\min_j  \lambda_j - \tilde{\lambda}_i $	$\ \mathbf{R}_i\ _2$	$q_i$
1-2	-7.61e-1±4.61e-1i	-7.61e-1±4.61e-1i	3.07e-8	1.69e+0	2.21
3-4	-1.33e+0±2.02e+0i	-1.33e+0±2.02e+0i	9.67e-8	1.38e+0	1.03
5-6	-1.57e-1±5.52e-1i	-1.57e-1±5.52e-1i	3.90e-9	2.67e-1	1.69
7-8	-8.29e-2±1.87e+0i	-8.29e-2±1.87e+0i	1.20e-9	1.03e-1	1.24
9-10	-2.19e-2±6.96e-1i	-2.19e-2±6.96e-1i	3.85e-10	3.86e-2	1.76
11-12	-2.22e-3±1.53e+0i	-2.22e-3±1.53e+0i	1.37e-10	1.05e-2	4.73
13-14	-9.98e-3±8.01e-1i	-9.98e-3±8.01e-1i	2.77e-10	8.71e-3	8.72e-1
15-16		1.58e-4±6.11e-1i	8.81e-2	2.84e-11	1.79e-7
17-18		-4.53e-5±9.04e-1i	1.03e-1	2.28e-12	5.04e-8
19-20		3.16e-5±8.55e-1i	5.48e-2	8.52e-13	2.69e-8

Table 4.3: Results for  $SNR = 80$ 

i	original poles $\lambda_i$	$\tilde{\lambda}_i$	$\min_j  \lambda_j - \tilde{\lambda}_i $	$\ \mathbf{R}_i\ _2$	$q_i$
1-2	-7.61e-1±4.61e-1i	-7.66e-1±4.53e-1i	9.46e-3	1.75	2.28
3-4	-1.33e+0±2.02e+0i	-1.32e+0±1.99e+0i	3.61e-2	1.38	1.04
5-6	-1.57e-1±5.52e-1i	-1.57e-1±5.53e-1i	1.61e-3	2.67e-1	1.69
7-8	-8.29e-2±1.87e+0i	-8.17e-2±1.87e+0i	2.68e-3	1.02e-1	1.25
9-10	-2.19e-2±6.96e-1i	-2.17e-2±6.97e-1i	5.06e-4	3.77e-2	1.73
11-12	-2.22e-3±1.53e+0i	-2.35e-3±1.53e+0i	1.54e-4	1.07e-2	4.58
13-14	-9.98e-3±8.01e-1i	-9.80e-3±8.01e-1i	2.13e-4	8.48e-3	8.65e-1
15-16		-3.72e-3±5.98e-1i	1.00e-1	6.46e-4	1.73e-1
17-18		3.92e-3±7.67e-1i	3.62e-2	3.67e-4	9.34e-2
19-20		-1.32e-4±7.91e-1i	1.36e-2	2.71e-4	2.03

Table 4.4: Results for  $SNR = 20$

Table 4.5 presents the errors after the SVD truncation of order 14, as well as after the 14 poles with the largest residues were chosen from a model of order 56.

Model	$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
Order 14 after SVD truncation	5.00e-1	4.39e-1
Order 56 after SVD truncation	8.02e-2	3.30e-2
Order 14 after largest residue selection	2.45e-2	2.44e-2
Noise	5.13e-2	2.69e-1

Table 4.5: Results for  $SNR = 20$

We looked at the pseudospectra of the original matrix and compared the location of the poles given by our improved algorithm for  $SNR = 20$  to the pseudospectra bounds corresponding to perturbations in the range  $10^{-1.6} - 10^{-2}$  (Fig. 4.3, generated using EigTool [63]). Pseudospectra can be defined as [64]

$$\Lambda_\epsilon(\mathbf{A}) = \{z \in \mathbb{C} : z \in \Lambda(\mathbf{A} + \mathbf{P}) \text{ for some } \mathbf{P} \text{ with } \|\mathbf{P}\| \leq \epsilon\} \quad (4.54)$$

The above definition is not for matrix pencils. Still, the poles recovered by selecting those with the largest residues of a high order system are enclosed by the circle in the complex plane which bounds the original poles, perturbed by adding matrices  $\mathbf{P}$  with norm up to  $10^{-2}$  to the matrix  $\mathbf{E}^{-1}\mathbf{A}$ .

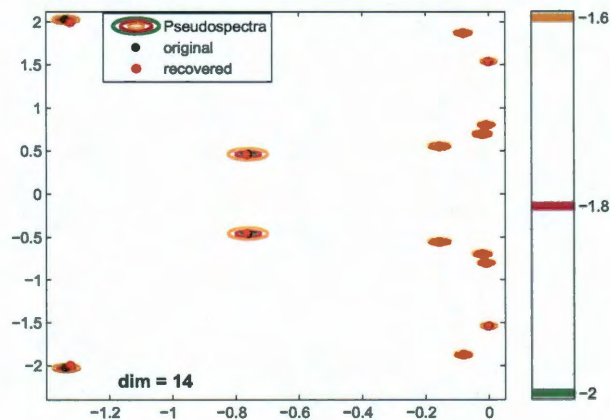


Figure 4.3: Pseudospectra of the  $\mathbf{A}$  matrix (contours correspond to levels of  $\epsilon$  of  $10^{-1.6}$ ,  $10^{-1.8}$ ,  $10^{-2}$ )

We performed Monte Carlo analyses for 20 different random perturbations for each SNR value. The  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  error results are shown in Fig. 4.4. We notice that, with a few exceptions, the  $\mathcal{H}_2$ -norm errors for our models are well below the values of the noise, while the  $\mathcal{H}_\infty$ -norm errors are close to those of the noise.

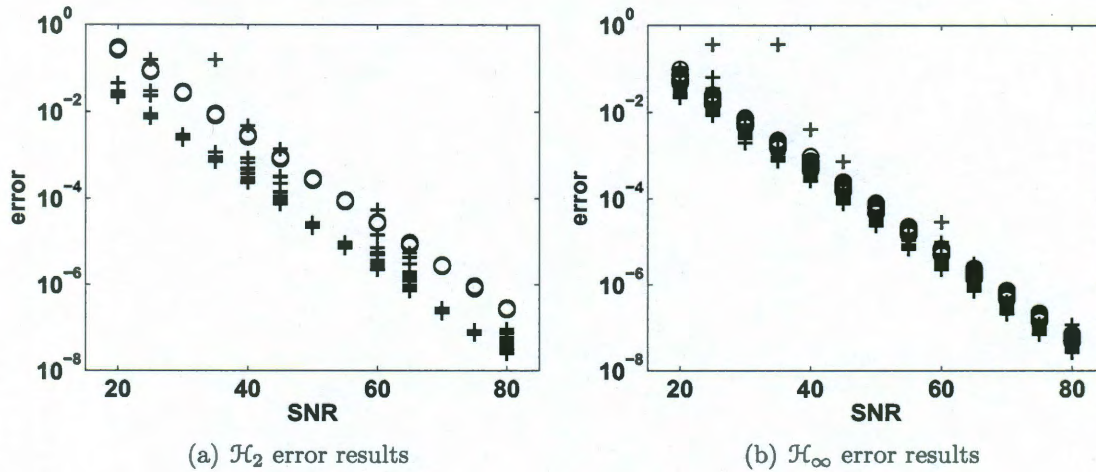


Figure 4.4: Results for different SNR values (the circles are the norms of the noise, while the pluses are the norms of our models)

## 4.4 Numerical Results

In this section, we apply the implementations proposed in Sect. 4.2 to various numerical examples. We often compare our approach to vector fitting (see details in Chapter 6). Our analysis focuses on the accuracy of the macromodels and the CPU time required to produce such models. We plot the singular values of the transfer function versus frequency of the model which provides the best trade-off between accuracy and CPU time, as well as the singular values of the measured frequency responses. The agreement in the singular values is an indication of the good quality of the model since, when comparing the magnitude and angle of some of the individual entries of the frequency response matrices to the model, we notice a very good match.

Experiments used column-wise VF with the same options:

- The starting poles are complex conjugate pairs with weak attenuation, distributed over the frequency band.
- The starting poles for each column are obtained by fitting the column sum with  $N_1 = 5$  iterations; these are used to fit the column itself with  $N_2 = 5$  iterations.
- The fast implementation of relaxed vector fitting [18, 20, 25] was used.

Experiments were performed on a Pentium Dual-Core at 2.2GHz with 3GB RAM.

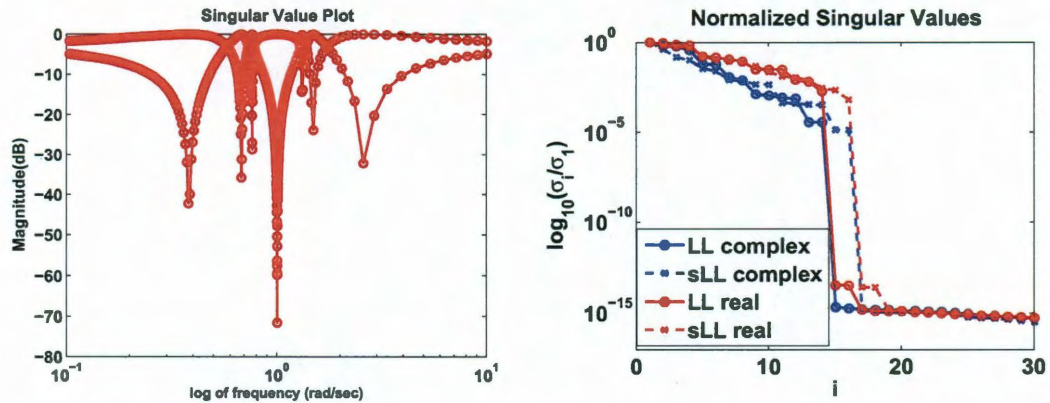
#### 4.4.1 Noise-free system with 2 ports, 14 poles and $\mathbf{D} \neq \mathbf{0}$

We consider an explicitly given system of order 14 with  $p = 2$  ports and a non-zero  $\mathbf{D}$ -term [47, 43]. This system has a priori known matrices and we create  $k = 608$  noise-free measurements of its transfer function on the imaginary axis between  $10^{-1}$  and  $10^1$  rad/sec. The purpose of this example is to show that our approaches can identify the system and recover the system matrices, but also to illustrate how the  $\mathbf{D}$  term affects the singular value drop of the Loewner pair.

Fig. 4.5(a) shows the magnitude of the original system, while Fig. 4.5(b) shows the normalized singular values of the Loewner and shifted Loewner matrices constructed using all samples in the complex and real approaches (we show the first 30 singular values as the rest are zero). The Loewner matrix has rank 14, while the shifted Loewner matrix has rank 16 so we generate models of order 16 with  $\mathbf{E}$  singular and  $\mathbf{A}$  invertible, to obtain the 14 finite poles and 2 infinite ones. Vector fitting was given  $N = 7$  starting poles and was required to produce a  $\mathbf{D}$  matrix.

Table 4.6 presents the CPU time and the errors for the resulting models. We conclude that all proposed approaches were able to identify the original system, while vector fitting did not (Fig. 4.6 and 4.7).

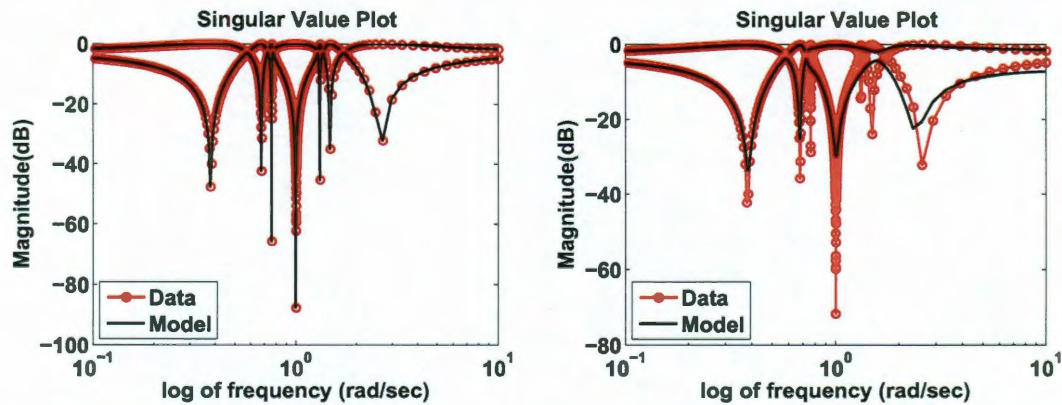
With  $k = 7$  starting poles, column-wise VF constructs a model of dimension  $n = p \cdot k = 2 \cdot 7 = 14$  (same order as in our approach), by fitting each of the two



(a) Frequency response of the original system (b) Singular value drop of the Loewner pencil  
 Figure 4.5: Original system and singular value drop of the Loewner pencil

Algorithm	CPU (s)	$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
<b>VF</b>	<b>1.15</b>	<b>1.0536</b>	<b>2.2278e-001</b>
SVD Complex	2.18	1.3937e-010	2.4269e-011
SVD Real	2.41	1.3146e-012	3.0687e-013
Adaptive Complex	0.42	2.0402e-009	2.5888e-010
<b>Adaptive Real</b>	<b>0.4</b>	<b>7.0155e-013</b>	<b>2.9316e-013</b>

Table 4.6: Results for an order 14 system with 2 ports



(a) Adaptive Real Approach builds a model which matches the data (b) VF does not provide a good model

Figure 4.6: Models for a system with  $p = 2$  ports



columns with the same poles. However, in this example, all entries of the transfer function share the same  $n = 14$  poles. With  $k = 14$  starting poles, each of the two columns are fitted by 14 poles and the resulting errors are similar to ours. Nonetheless, overmodeling occurs as the realization has order  $n = 28$  and each pole has multiplicity 2, so an additional step [19] is needed to recover the original system.

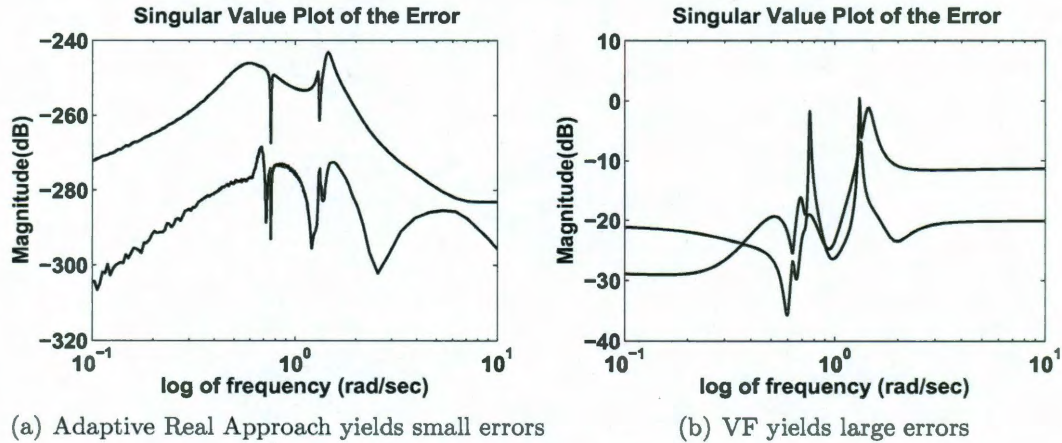


Figure 4.7: Error plots for a system with  $p = 2$  ports

#### 4.4.2 S-parameter measurements

These scattering (S-)parameter measurements were performed using a vector network analyzer (VNA) and were provided by CST AG [43]. This set contains  $N = 100$  frequency samples between 10MHz and 1GHz from a device with 50 ports. For better conditioning, all frequencies were scaled by  $10^{-6}$ .

Fig. 4.8(a) shows the normalized singular values of the Loewner and shifted Loewner matrices constructed from all measurements, using the complex and real approaches. The singular values of the Loewner matrices show a decay of several orders of magnitude between the 9th and 10th, while the singular values of the shifted Loewner matrices show a decay of several orders of magnitude between the 59th and 60th. The drop in singular values suggests that there is an underlying  $\mathbf{D}$ -term and, moreover, allows to identify the order of the system.

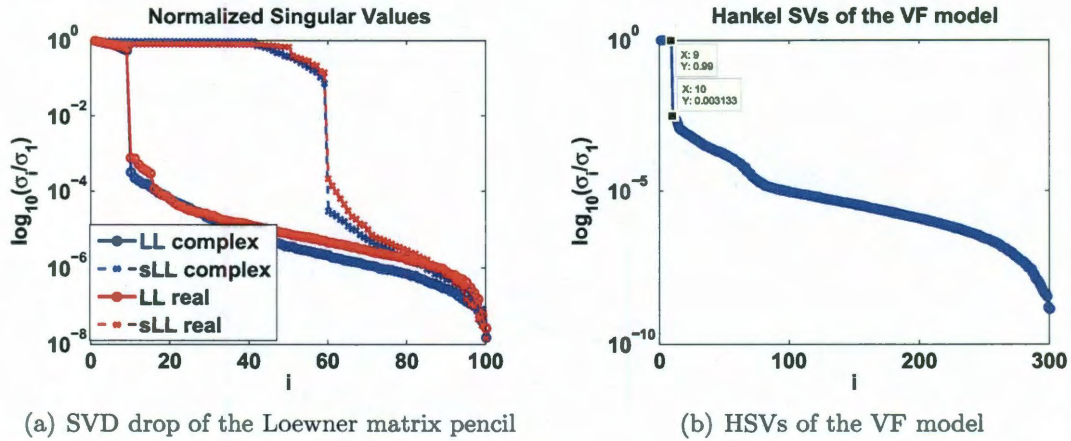


Figure 4.8: Singular value drop of the Loewner matrix pencil and Hankel singular value drop of the VF model

Table 4.7 presents the results for models of order  $n = 9$  obtained with our approaches, after extracting the  $\mathbf{D}$  matrix. Vector fitting was required to produce an asymptotic  $\mathbf{D}$  matrix, but the lowest order model we could construct was  $n = 100$ . All proposed algorithms yield better fits than VF.

For comparable errors, VF needs to build a model of order  $n = 300$ . Fig. 4.8(b) shows the Hankel singular values of this large VF model, which exhibit a decay of two orders of magnitude between the 9th and 10th singular values. After reducing to order  $n = 9$  using balanced truncation (BT) [23], the resulting errors are close to those of our models. The reason behind the poor performance of VF is that each column is fitted by the same poles in the column-wise VF implementation, so for the  $n = 100$  model, each column shares only two ( $\frac{n}{p} = \frac{100}{50} = 2$ ) poles, which is too restrictive.

The model obtained with the complex SVD approach is shown in Fig. 4.9(a), while the VF model of order 100 is shown in Fig. 4.9(b). The plots in Fig. 4.10(a) and 4.10(b) show the singular values of the error matrices (defined in Eq.(3.7)).

Fig. 4.11 compares the measured  $S_{1,1}$  and  $S_{10,20}$  entries to the model obtained with the real SVD approach and with VF. Our model is very close to the data.

Table 4.8 shows that all our models are stable and very close to being passive, as

Algorithm	CPU time (s)	$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
<b>VF (n=100)</b>	<b>1.97</b>	<b>3.4831e-001</b>	<b>4.8449e-002</b>
<b>SVD Complex (n=9)</b>	<b>0.12</b>	<b>5.3638e-003</b>	<b>5.2756e-004</b>
SVD Real (n=9)	0.15	6.1575e-003	5.9957e-004
Adaptive Complex (n=9)	1.21	7.0829e-003	8.3487e-004
Adaptive Real (n=9)	1.01	5.3318e-003	4.4008e-004
VF (n=300)	5.27	4.2894e-003	2.4770e-004
VF & BT (n=9)	7.62	7.3339e-003	1.4120e-003

Table 4.7: Results for a device with  $p = 50$  ports

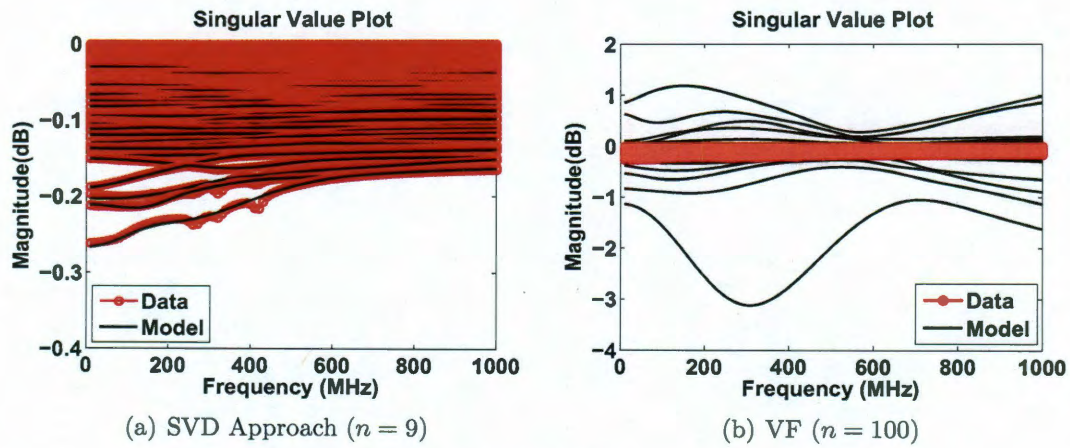


Figure 4.9: Models for a device with  $p = 50$  ports

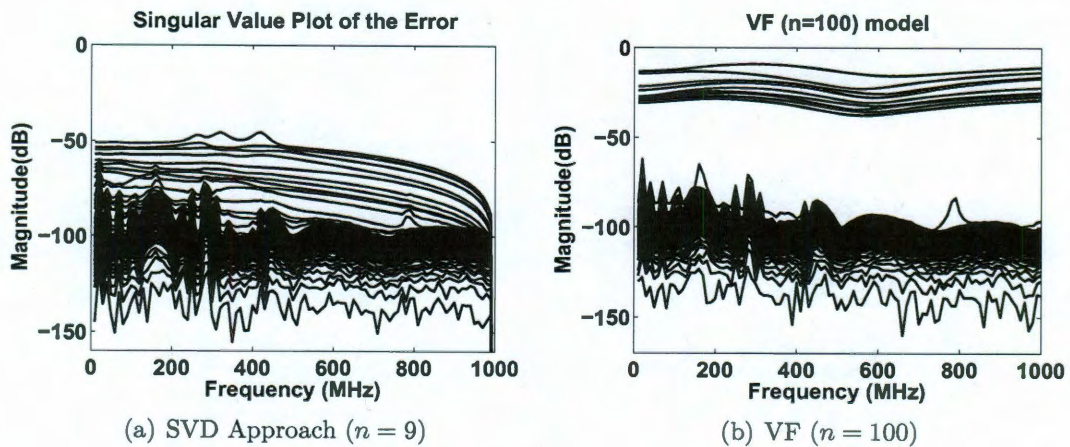


Figure 4.10: Error plots for a device with  $p = 50$  ports

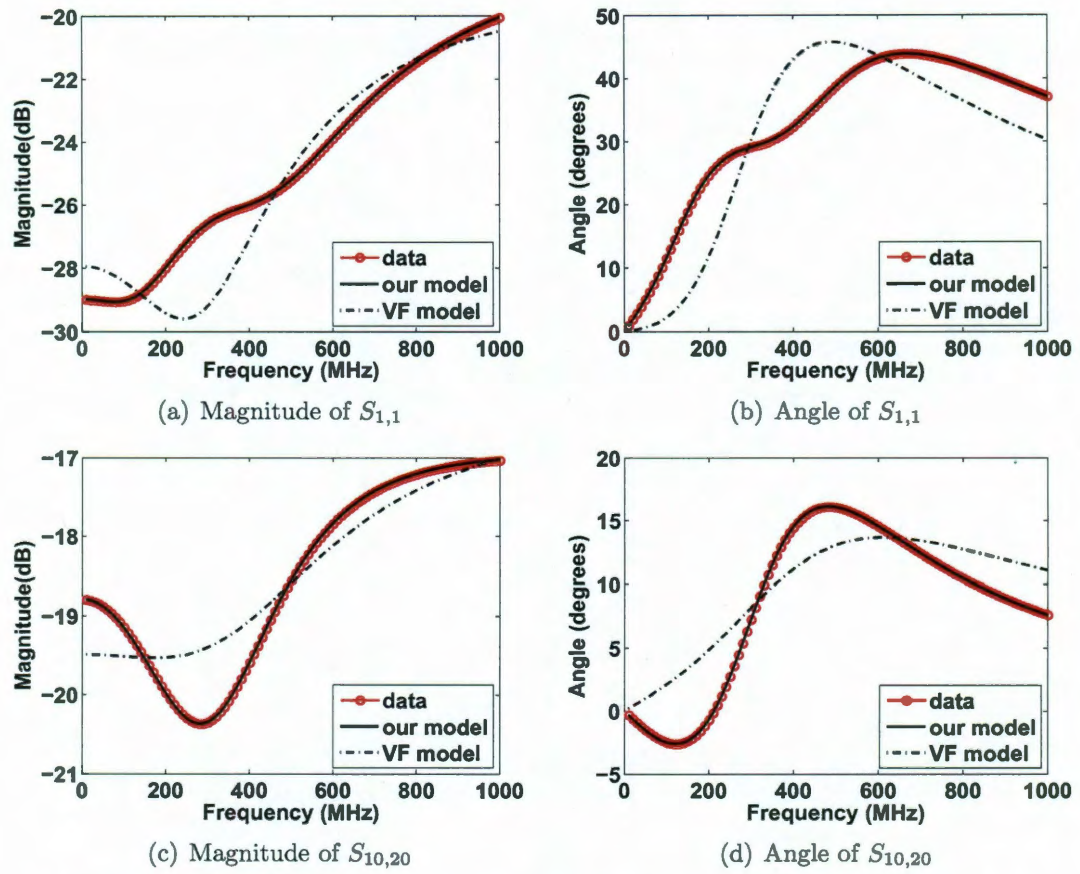


Figure 4.11: Modeling S-parameter entries for a device with  $p = 50$  ports

their  $\mathcal{H}_\infty$ -norm is at most 0.2% above one, so by dividing the  $\mathbf{B}$  or the  $\mathbf{C}$  matrix of the corresponding realization by this  $\mathcal{H}_\infty$ -norm, one will introduce an error of  $-50\text{dB}$ , but the resulting systems will be bounded-real passive.

Algorithm	Stable	$\mathcal{H}_\infty$ -norm
VF ( $n = 100$ )	Yes	1.9729
SVD Complex	Yes	1.0002
SVD Real	Yes	1.0002
Adaptive Complex	Yes	1.0003
Adaptive Real	Yes	1.0002
VF ( $n = 300$ )	Yes	1.9761
VF & BT ( $n = 9$ )	Yes	1.9763

Table 4.8: Stability and passivity results for a device with  $p = 50$  ports

### 4.4.3 RC transmission line

We consider an RC transmission line [65] with 3252 voltage nodes (which are taken as states in the MNA formulation) and 21 of them are considered as terminals (i.e., inputs/outputs):

$$(\mathbf{G} + s\mathbf{C})\mathbf{x}(s) = \mathbf{B}\mathbf{u}(s)$$

where the MNA matrices  $\mathbf{G}$  and  $\mathbf{C}$  are symmetric, non-negative definite and sparse. The conductance matrix  $\mathbf{G}$  contains the values of the resistors, while the capacitance matrix  $\mathbf{C}$  contains the values of the capacitors. The states  $\mathbf{x}(t)$  are the node voltages and  $\mathbf{u}(t)$  are the currents injected into the terminals. The outputs are taken as the voltage drops at the terminal nodes:  $\mathbf{y}(s) = \mathbf{B}^T\mathbf{x}(s)$ , so the impedance transfer function is  $\mathbf{H}(s) = \mathbf{B}^T(\mathbf{G} + s\mathbf{C})^{-1}\mathbf{B}$ .

We consider 30 admittance (Y-)parameter measurements logarithmically distributed between  $10^0$  and  $10^{15}$  and build the Loewner matrix pencil using matrix data (each  $21 \times 21$  block is  $\mathbb{L}_{i,j} = \frac{1}{\mu_j - \lambda_i}(\mathbf{V}_j - \mathbf{W}_i)$  and  $\sigma\mathbb{L}_{i,j} = \frac{1}{\mu_j - \lambda_i}(\mu_j\mathbf{V}_j - \lambda_i\mathbf{W}_i)$ , respectively, for  $i = 1, \dots, 30$ ,  $j = 1, \dots, 30$ ,  $i \neq j$ ). To preserve the symmetry of the original problem, the left and right data are taken equal to each other:  $\mathbf{V}_i = \mathbf{W}_i$  and

$\mu_i = \lambda_i$ , for  $i = 1, \dots, 30$ . Since the entries in the Loewner pencil are not defined for  $i = j$ , we make use of the derivative measurement:

$$\mathbf{H}'(s) = -\mathbf{B}^T (\mathbf{G} + s\mathbf{C})^{-1} (-\mathbf{C}) (\mathbf{G} + s\mathbf{C})^{-1} \mathbf{B}.$$

Therefore,  $\mathbb{L}_{ii} = \mathbf{H}'(\lambda_i)$ ,  $i = 1, \dots, 30$  and  $\sigma\mathbb{L}_{ii} = \mathbf{H}(\lambda_i) + \lambda_i\mathbf{H}'(\lambda_i)$ . The Loewner and shifted Loewner matrices are of dimension  $30 \times 21 = 630$ . The singular value drop of the matrix pencil constructed from the measurements is shown in Fig. 4.12.

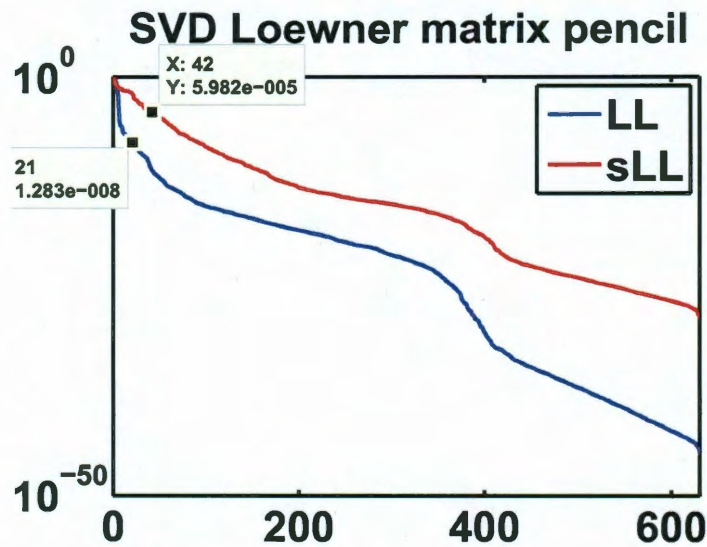


Figure 4.12: Singular value drop of the Loewner matrix pencil

We truncate to orders 21 and 42. The fact that the singular values of the shifted Loewner matrix decay slower than those of the Loewner matrix indicate that there exists an underlying non-zero  $\mathbf{D}$ -term (see Sect. 4.1.4). Fig. 4.13 and 4.14 show the largest and smallest singular values of the  $\mathbf{Y}$ -parameter measurements, as well as the largest and smallest singular values of the transfer function matrix obtained by evaluating the reduced model for the same frequency samples considered previously.

Fig. 4.15 shows the AC analysis of the original and the two reduced circuits. The netlist representation was constructed based on RLCSYN unstamping [66], [67] and the re-simulation was performed with Spectre [68], courtesy of NXP Semiconductors.

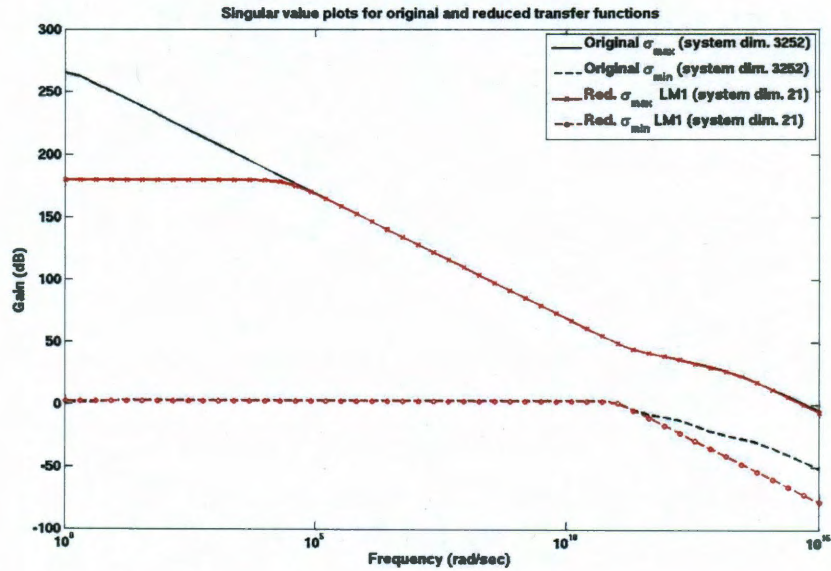


Figure 4.13: Largest and smallest singular values of the Y-parameter measurements versus the reduced model

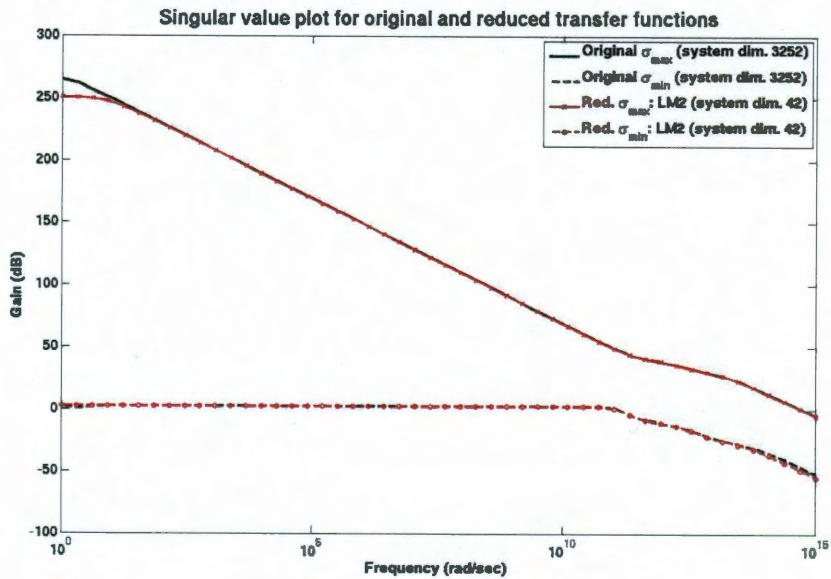


Figure 4.14: Largest and smallest singular values of the Y-parameter measurements versus the reduced model

The order 42 model yields a signal which is very close to the original.

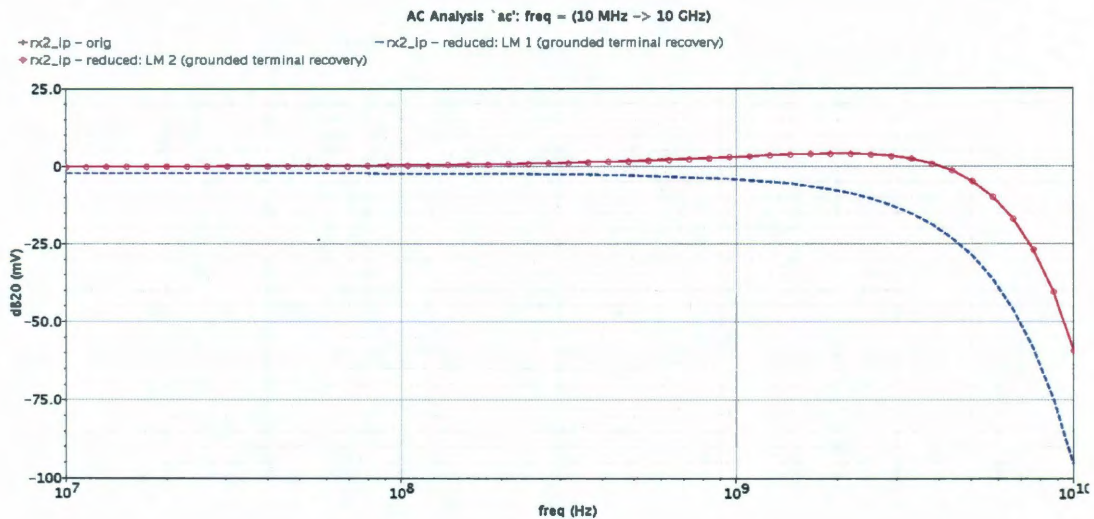


Figure 4.15: AC simulation of the original (red, 3253 nodes), and the two reduced netlists: order 22 (blue) and order 43 (magenta)

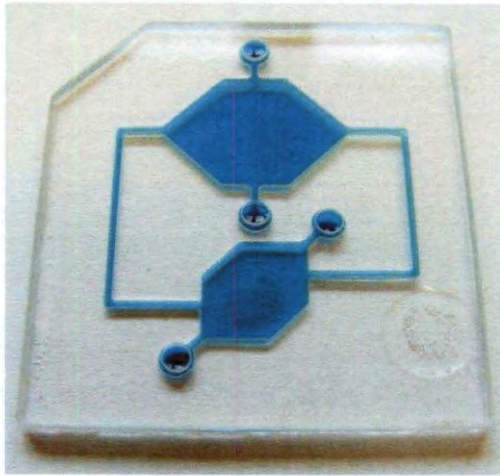
Fig. 4.13, 4.14 and 4.15 represent post-processing steps and are also included in [69], but the reduced system was obtained by the author of the present thesis.

#### 4.4.4 Biochip

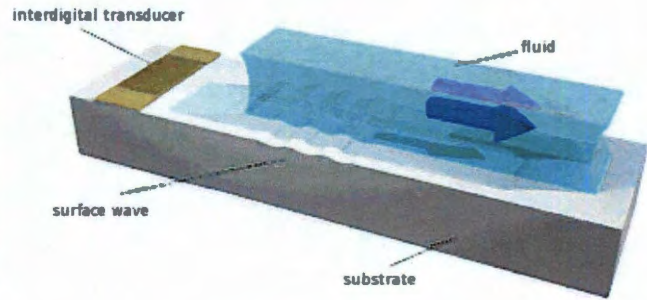
Microfluidic systems represent a promising approach towards the realization of a so called 'lab-on-a-chip', a device made by integrating several lab functions on a chip. Such a device dealing with microfluids is called a *microfluidic biochip* (Fig. 4.16). These are used in pharmaceutical, medical, and forensic applications for genotyping in genomics, protein profiling in proteomics, and cytometry in cell analysis.

The compressible Stokes equations with low viscosity coefficient  $\nu$  [70, 71, 72] are





(a) Picture from [70, 71]



(b) Picture from [72]

Figure 4.16: Microfluidic Biochip

used to model the biochip [70, 71, 72]. The resulting state space is

$$\begin{bmatrix} \mathbf{M}_v \\ \mathbf{M}_p \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} -\nu \mathbf{A} & -\mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}(t) \\ p(t) \end{bmatrix} + \begin{bmatrix} \mathbf{K} \\ \mathbf{L} \end{bmatrix} \mathbf{h}(t) \quad (4.55)$$

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}(t) \\ p(t) \end{bmatrix} \quad (4.56)$$

with  $\mathbf{v}(0) = \mathbf{0}$  and  $p(0) = 0$  for  $t \in (0, T)$ . The state variable  $\mathbf{v}(t)$  denotes flow velocity, while  $p(t)$  is the pressure. The matrices  $\mathbf{M}_v$  and  $\mathbf{M}_p$  are the mass matrices corresponding to velocity and pressure, respectively,  $\mathbf{A}$  is the stiffness matrix,  $\mathbf{B}$  is the divergence matrix while  $\mathbf{K}$  and  $\mathbf{L}$  are the input matrices. The output  $\mathbf{y}(t)$  is the integral of the curl over the entire spatial domain  $\Omega$ .

The low viscosity coefficient  $\nu$  puts most of the eigenvalues of the system close to zero (see Fig. 4.18). Note that for high viscosity, namely  $\nu = 1$ , all eigenvalues are real and the system would be easy to approximate due to the fact that the response would contain no resonant peaks. One can easily note that the poles for  $\nu = 10^{-6}$  are closer to the imaginary axis (within  $10^{-3}$ ) than for  $\nu = 10^{-2}$ .

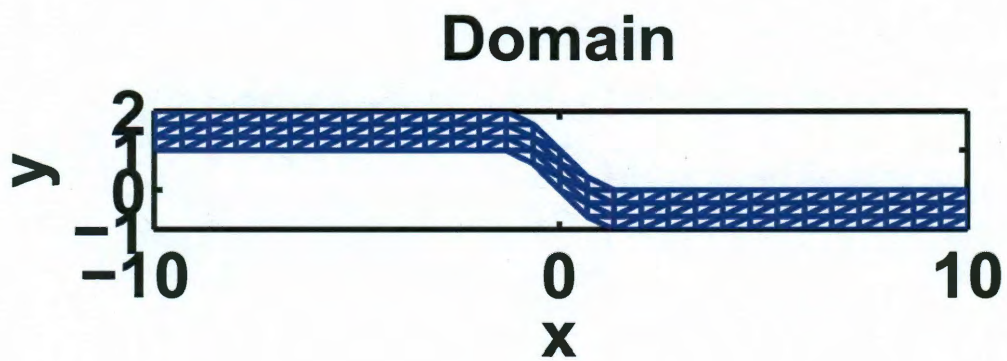


Figure 4.17: Discretization of the spatial domain  $\Omega$  for modeling the biochip

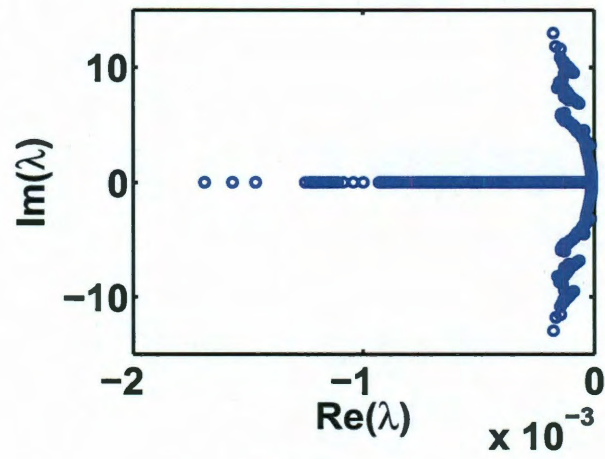
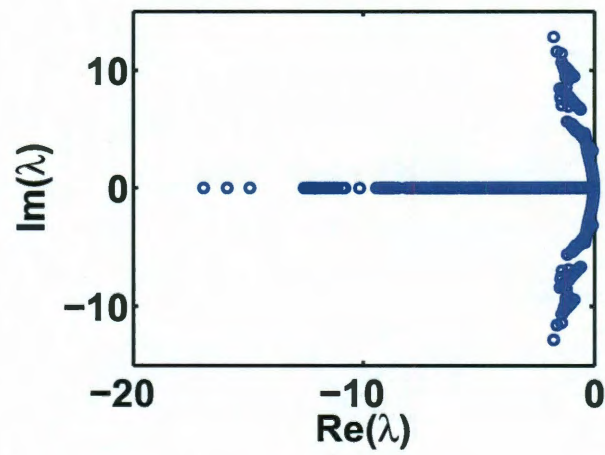
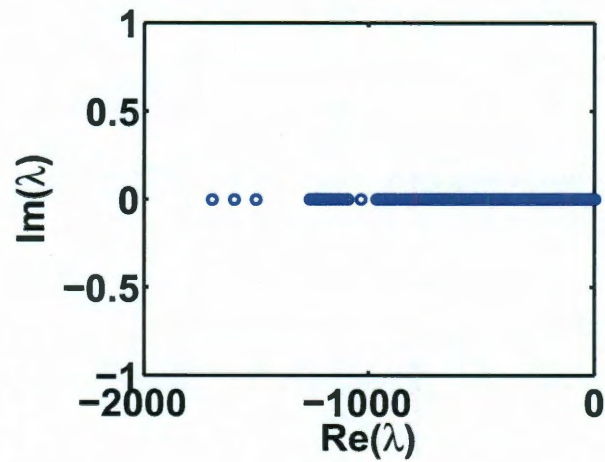
(a)  $\nu = 10^{-6}$ (b)  $\nu = .01$ (c)  $\nu = 1$ 

Figure 4.18: Poles for various values of  $\nu$ ; the smaller the viscosity value, the closer the poles are to the imaginary axis

From this point on, we consider the viscosity to be  $\nu = 10^{-6}$ . We use an “original” system of dimension  $N = 724$  obtained by discretizing the PDE using the Finite Element Method (Fig. 4.17) with P2-P1 elements, i.e., continuous piecewise quadratic elements for velocity and piecewise linear elements for pressure.

We compare reduced systems of order  $n = 40$  obtained with Balanced Truncation to those obtained by applying the Loewner matrix framework to the  $N_s = 1000$  frequency response measurements considered for frequencies varying logarithmically between  $10^{-2.2}$  and  $10^1$ . Fig. 4.19(a) shows the magnitude plot of the original system and a reduced one of order 40. The behavior of the original system is only shown at the samples that were considered for building the Loewner matrix pencil and the reduced system is evaluated precisely at these samples. The response of the original system is approximated quite well by the reduced system for low frequencies. Moreover, the peaks with the largest magnitude are matched.

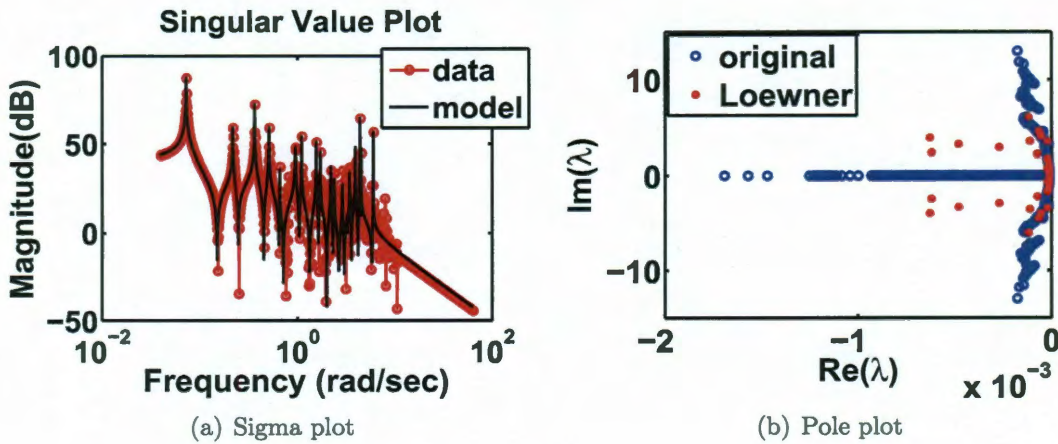


Figure 4.19: Sigma magnitude plot and pole plot comparing the original and reduced order 40 system

Fig. 4.19(b) shows the poles of the original and the order 40 reduced system. As expected from the magnitude plot, some of the poles close to the real axis (those with low frequency) and close to the imaginary axis (those corresponding to the highest resonant peaks) are matched.

Fig. 4.20(a) shows the magnitude plot of the original system and the reduced order  $n = 50$  system. The response of the original system is approximated better by this reduced system than the one of order  $n = 40$ .

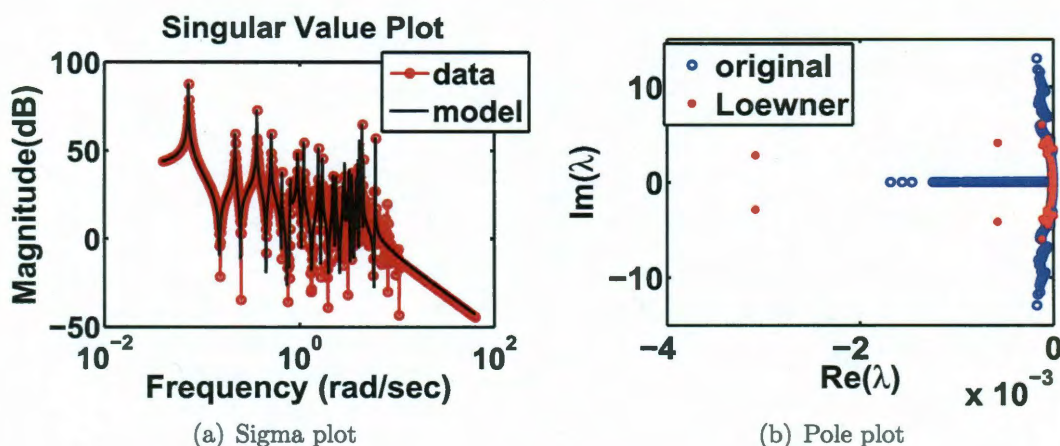


Figure 4.20: Sigma magnitude plot and pole plot comparing the original and reduced order 50 system

Fig. 4.20(b) shows the poles of the original and order 50 reduced system. Like in the case of the order  $n = 40$  reduced system, some of the original poles are matched by this reduced model, namely those close to the real axis (corresponding to the peaks at low frequency) and close to the imaginary axis (corresponding to the resonant peaks with the largest magnitude).

Fig 4.21 shows a comparison of the time responses of the original and reduced systems (the plots show the output obtained for a sinusoidal time domain input signal). The order  $n = 40$  reduced systems are able to track the signal only up to about 2s (see Fig. 4.21(a)). The reduced systems of order  $n = 50$  shown in 4.21(b) are much better, with our reduced system tracking the original response better for small times and the BT reduced system tracking better after approx. 7.5s.

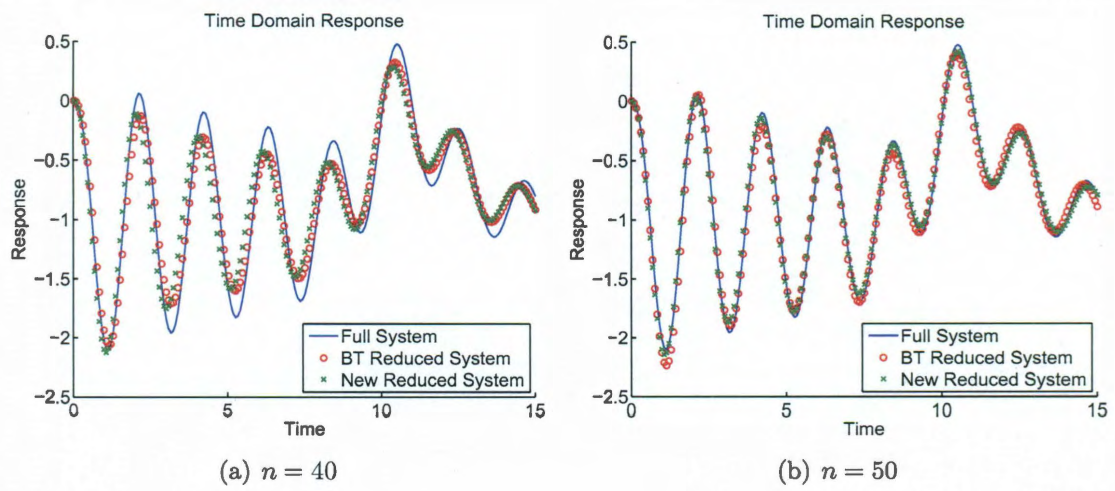


Figure 4.21: Comparison of time domain responses

## Parametric models

---

Another topic addressed in this thesis is the problem of generating parametric models from measurements performed with respect to the frequency, but also with respect to one or more design parameters. These design parameters could relate to geometry (the width, length or height of some components), material properties (the electrical conductivity or dielectric constant of the various materials which can be used) or varying boundary conditions (of Dirichlet, Neumann or Robin type). These models are suited for optimizing certain performance criteria over the space of design variables.

Fig. 5.1 shows the variation in the step response of the RLC circuit as the value of the inductor  $L_2$  increases. The transfer function for this circuit (considering the voltage as input and the current as output), when assuming the value of the inductor  $L_2$  to be a design parameter and the rest of the circuit elements are unit-valued, depends on two variables: the Laplace variable,  $s$ , and the inductor value:

$$\mathbf{H}(s, L_2) = \frac{L_2 s^3 + L_2 s^2 + 2s + 1}{s^4 L_2 + 2s^3 L_2 + 3s^2 L_2 + 2s^2 + sL_2 + 3s + 1}.$$

Therefore, the approach presented in this thesis to address the parametric macro-modeling problem consists of building  $2D$  transfer functions, in other words, transfer functions depending on two variables. Using the frequency response measurements

obtained for several parameter values, we generalize the *Loewner matrix* to the two variable case to construct models which are reduced both with respect to frequency and to the parameter (in Sect.5.1 - 5.5).

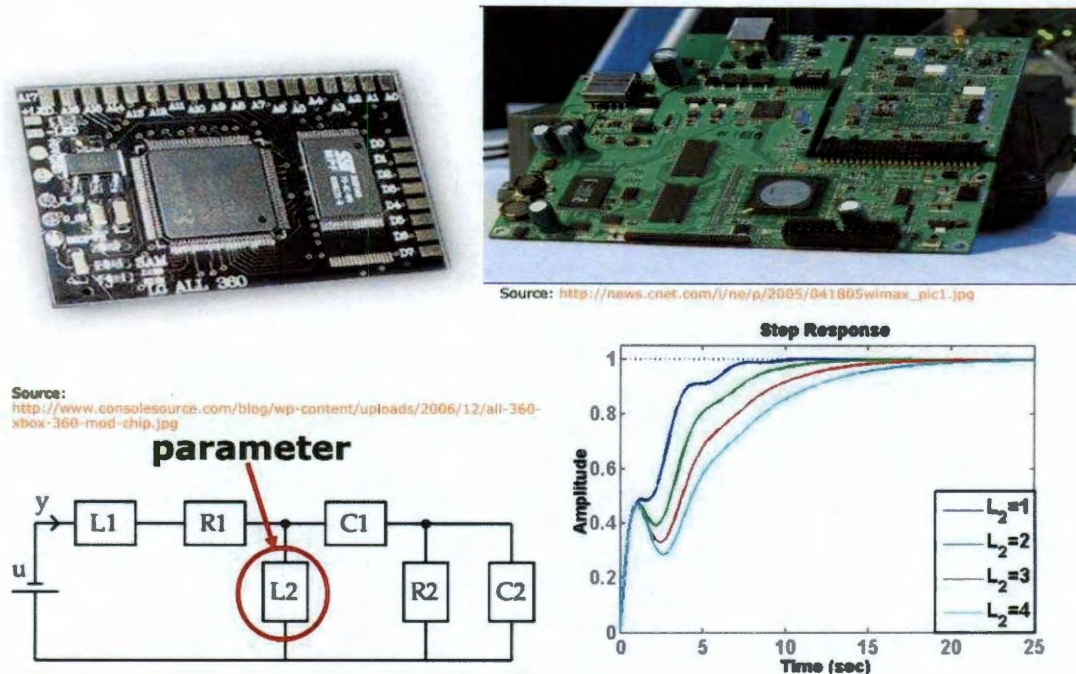


Figure 5.1: Motivation for using parametric models

## 5.1 The Loewner matrix in the two-variable case

We denote the space of polynomials in two variables,  $s$  and  $t$ , of degree at most  $n$  in  $s$  and at most  $m$  in  $t$  by  $\mathcal{P}_{n,m}$ . For this linear space of dimension  $(n+1)(m+1)$ , one can consider different bases. The *monomial basis* consists of all monomials of the form  $s^i t^j$ ,  $i=0, 1, \dots, n$ ,  $j=0, 1, \dots, m$ . This basis for the two variable case can be regarded as the Kronecker product between the monomial bases of the space of polynomials  $\mathcal{P}_n$  in  $s$  and of the space of polynomials  $\mathcal{P}_m$  in  $t$ . An alternative basis which is better suited for generalizing the Loewner matrix to two variables is the *Lagrange basis* [51, 73]. This is constructed using given (distinct) complex numbers



$\lambda_i, i=0, \dots, n$ , and  $\pi_j, j=0, \dots, m$ , by defining the *Lagrange* polynomials

$$\mathbf{q}_{i,j}(s, t) := \prod_{i' \neq i} \frac{(s - \lambda_{i'})}{(\lambda_i - \lambda_{i'})} \prod_{j' \neq j} \frac{(t - \pi_{j'})}{(\pi_j - \pi_{j'})}. \quad (5.1)$$

For arbitrary  $\alpha_{i,j}$ , the polynomial

$$\mathbf{p}(s, t) := \sum_{i=0}^n \sum_{j=0}^m \alpha_{i,j} \mathbf{q}_{i,j}(s, t), \quad (5.2)$$

is the *two-variable Lagrange form of the interpolating polynomial* that satisfies the interpolation conditions  $\mathbf{p}(\lambda_i, \pi_j) = \alpha_{i,j}$ ,  $i=0, \dots, n$ ,  $j=0, \dots, m$ . This is the unique polynomial in  $\mathcal{P}_{n,m}$  satisfying these interpolation conditions.

We can express any rational function of degree  $n$  in  $s$  and  $m$  in  $t$  as a ratio of two polynomials (in particular Lagrange polynomials):

$$\mathbf{g}(s, t) = \frac{\sum_{i=0}^n \sum_{j=0}^m \beta_{i,j} \mathbf{q}_{i,j}(s, t)}{\sum_{i=0}^n \sum_{j=0}^m \alpha_{i,j} \mathbf{q}_{i,j}(s, t)}. \quad (5.3)$$

Assuming  $\alpha_{i,j} \neq 0, \forall i, j$ , we set  $(n+1)(m+1)$  interpolation conditions:  $\mathbf{g}(\lambda_i, \pi_j) = \frac{\beta_{i,j}}{\alpha_{i,j}} =: \mathbf{w}_{i,j}$ . Since

$$\mathbf{g}(s, t) = \frac{\sum_{i=0}^n \sum_{j=0}^m \frac{\beta_{i,j}}{(s - \lambda_i)(t - \pi_j)}}{\sum_{i=0}^n \sum_{j=0}^m \frac{\alpha_{i,j}}{(s - \lambda_i)(t - \pi_j)}}, \quad (5.4)$$

it follows that  $\mathbf{g}$  satisfies the equation

$$\sum_{i=0}^n \sum_{j=0}^m \alpha_{i,j} \frac{\mathbf{g} - \mathbf{w}_{i,j}}{(s - \lambda_i)(t - \pi_j)} = 0. \quad (5.5)$$

To determine  $\alpha_{i,j}$ ,  $(n+1)(m+1)$  additional conditions are imposed on  $\mathbf{g}$ :

$$\mathbf{g}(\mu_i, \nu_j) = \mathbf{v}_{i,j}, \quad i = 0, \dots, n, \quad j = 0, \dots, m, \quad (5.6)$$

where  $(\mu_i, \nu_j; \mathbf{v}_{i,j})$ , are given triples of complex numbers ( $\lambda_i \neq \mu_j, \pi_i \neq \nu_j, i \neq j, \forall i, j$ ).

Using the notation  $\ell_{i,j}^{k,l} = \frac{\mathbf{v}_{k,l} - \mathbf{w}_{i,j}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)}$  and substituting Eq. (5.6) in Eq. (5.5), we

obtain the condition  $\mathbb{L}\mathbf{c} = \mathbf{0}$  for determining the coefficients  $\alpha_{i,j}$ , where

$$\mathbb{L} = \begin{bmatrix} \ell_{0,0}^{0,0} & \cdots & \ell_{0,m}^{0,0} & \ell_{1,0}^{0,0} & \cdots & \ell_{1,m}^{0,0} & \cdots & \ell_{n,0}^{0,0} & \cdots & \ell_{n,m}^{0,0} \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \ell_{0,0}^{0,m} & \cdots & \ell_{0,m}^{0,m} & \ell_{1,0}^{0,m} & \cdots & \ell_{1,m}^{0,m} & \cdots & \ell_{n,0}^{0,m} & \cdots & \ell_{n,m}^{0,m} \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \ell_{0,0}^{n,0} & \cdots & \ell_{0,m}^{n,0} & \ell_{1,0}^{n,0} & \cdots & \ell_{1,m}^{n,0} & \cdots & \ell_{n,0}^{n,0} & \cdots & \ell_{n,m}^{n,0} \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \ell_{0,0}^{n,m} & \cdots & \ell_{0,m}^{n,m} & \ell_{1,0}^{n,m} & \cdots & \ell_{1,m}^{n,m} & \cdots & \ell_{n,0}^{n,m} & \cdots & \ell_{n,m}^{n,m} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \alpha_{0,0} \\ \vdots \\ \alpha_{0,m} \\ \alpha_{1,0} \\ \vdots \\ \alpha_{1,m} \\ \vdots \\ \alpha_{n,0} \\ \vdots \\ \alpha_{n,m} \end{bmatrix}. \quad (5.7)$$

The matrix  $\mathbb{L}$  defined above is the *two-variable Loewner matrix* associated with the column and row data sets

$$\begin{aligned} P_c &= \{(\lambda_i, \pi_j; \mathbf{w}_{i,j}) : i=1, \dots, n', j=1, \dots, m'\}, \\ P_r &= \{(\mu_k, \nu_l; \mathbf{v}_{k,l}) : k=1, \dots, n', l=1, \dots, m'\}, \end{aligned} \quad (5.8)$$

where  $n' = n + 1$ ,  $m' = m + 1$ . It is a square matrix of dimension  $(n + 1)(m + 1) \times (n + 1)(m + 1)$ . Thus,  $\mathbf{g}$  satisfies the additional interpolation constraints if the vector  $\mathbf{c}$ , of dimension  $(n + 1)(m + 1)$ , is in the right kernel of  $\mathbb{L}$ .

**Lemma 5.1.1.** *Given the rational function  $\mathbf{g}$  of degree  $(m, n)$  and the arrays in Eq. (5.8), assuming  $n' \geq n + 1$ ,  $m' \geq m + 1$ , then the rank of the Loewner matrix  $\hat{\mathbb{L}}$  is  $n'm' - (n' - n)(m' - m)$ . Moreover, let  $\hat{\mathbf{c}}^T = [\hat{\alpha}_{1,1} \dots \hat{\alpha}_{1,m'} \dots \hat{\alpha}_{n',1} \dots \hat{\alpha}_{n',m'}] \in \mathbb{C}^{n'm'}$  satisfy  $\hat{\mathbb{L}}\hat{\mathbf{c}} = \mathbf{0}$  and  $\hat{\mathbf{g}}$  be defined by means of  $\sum_{i=0}^{n'} \sum_{j=0}^{m'} \hat{\alpha}_{i,j} \frac{\hat{\mathbf{g}} - \mathbf{w}_{i,j}}{(s - \lambda_i)(t - \pi_j)} = \mathbf{0}$ . Then  $\hat{\mathbf{g}} = \mathbf{g}$ .*

**Remark 5.1.1.** *In the one variable case, the rank of the Loewner matrix is independent of the number of measurements. This nice property not longer holds in the two-variable case.*

In a real applications scenario, we are given two-variable interpolation data and we wish to construct a rational function using the two-variable Loewner matrix. First, we determine the degree of the underlying rational function. We seek integers  $(n, m)$  to write  $\mathbf{g}$  as the quotient of two polynomials  $\mathbf{n}$ ,  $\mathbf{d}$ , both belonging to  $\mathcal{P}_{n,m}$ . By fixing  $t$  and computing the rank of the Loewner matrix constructed for varying  $s$  we determine  $n$ , and similarly, by fixing  $s$  and computing the rank of the Loewner matrix constructed for varying  $t$  we determine  $m$ . Next, we construct an interpolant using the Loewner matrix in Eq. (5.7). We recover the underlying  $\mathbf{g}$  from Eq. (5.3), where  $\mathbf{c}$  defined in Eq. (5.7) satisfies  $\mathbf{L}\mathbf{c}=\mathbf{0}$ , and  $\beta_{i,j}=\alpha_{i,j}\mathbf{w}_{i,j}$ .

## 5.2 Generalized state-space realizations

In this section we derive generalized state-space (descriptor) realizations in the Loewner matrix framework [51].

### 5.2.1 Single variable realizations

Of all descriptor realizations for the one variable rational function  $\mathbf{g}(s) = \frac{\sum_{i=0}^n b_i s^i}{\sum_{i=0}^n a_i s^i}$ , with  $b_n \neq 0$  or  $a_n \neq 0$  or both, we present one that can be generalized to the two variable case, where the corresponding dimension is low [51, 73].

We introduce the following notation

$$\mathbf{J}(x; k) = \begin{bmatrix} x & -1 & & & \\ & x & -1 & & \\ & & \ddots & \ddots & \\ & & & x & -1 \end{bmatrix} \in \mathbb{R}^{k \times (k+1)}[x], \quad (5.9)$$

and  $\mathbf{a} = [a_0, a_1, \dots, a_n]$ ,  $\mathbf{b} = [b_0, b_1, \dots, b_n] \in \mathbb{R}^{1 \times (n+1)}$ , while  $\mathbf{e}_k$  denotes the  $k^{\text{th}}$  canonical unit vector of dimension  $n+1$ .

**Lemma 5.2.1.** *The triplet  $\widehat{\Phi}$  (the inverse of the resolvent),  $\widehat{\mathbf{B}}$  and  $\widehat{\mathbf{C}}$  constitutes a descriptor realization of  $\mathbf{g}$ :*

$$\mathbf{g}(s) = \widehat{\mathbf{C}}\widehat{\Phi}^{-1}(s)\widehat{\mathbf{B}}, \quad \widehat{\mathbf{C}} = \mathbf{b}, \quad \widehat{\Phi}(s) = \begin{bmatrix} \mathbf{J}(s; n) \\ \mathbf{a} \end{bmatrix}, \quad \widehat{\mathbf{B}} = \mathbf{e}_{n+1}, \quad (5.10)$$

of dimension  $n+1$  which is  $R$ -controllable and  $R$ -observable [74], that is  $[\widehat{\Phi}(s), \widehat{\mathbf{B}}]$  and  $[\widehat{\Phi}^*(s), \widehat{\mathbf{C}}^*]$  (\* denotes transpose for real matrices and complex conjugate transpose for complex matrices) have full rank  $\forall s \in \mathbb{C}$ .

For  $\mathbf{g}$  expressed in a Lagrange basis, the realization is as follows. We define the Lagrange basis analogue of Eq. (5.9):

$$\mathbf{J}_{\text{lag}}(x; \lambda_i, k) = \begin{bmatrix} x - \lambda_1 & \lambda_2 - x & & & \\ x - \lambda_1 & 0 & \lambda_3 - x & & \\ \vdots & & \ddots & \ddots & \\ x - \lambda_1 & & & 0 & \lambda_{k+1} - x \end{bmatrix}, \quad (5.11)$$

and  $\mathbf{a} = [\alpha_0, \alpha_1, \dots, \alpha_n] = \mathbf{c}^T$  (where  $\mathbf{c}$  is the right null space of  $\mathbb{L}$ ), while  $\mathbf{b} = [\beta_0, \beta_1, \dots, \beta_n] \in \mathbb{R}^{1 \times (n+1)}$  (where  $\beta_i = \alpha_i \mathbf{w}_i$ ).

**Lemma 5.2.2.** *The following is a descriptor realization of  $\mathbf{g}$ :*

$$\widehat{\mathbf{C}} = \mathbf{b}, \quad \widehat{\Phi}(s) = \begin{bmatrix} \mathbf{J}_{\text{lag}}(s; \lambda_i, n) \\ \mathbf{a} \end{bmatrix}, \quad \widehat{\mathbf{B}} = \mathbf{e}_{n+1}, \quad (5.12)$$

of dimension  $n+1$  which can represent arbitrary rational functions, including polynomials. Also, it is  $R$ -controllable and  $R$ -observable.

The following corollary is useful for the two-variable case.

**Corollary 5.2.1.** *With the notation as in Lemmas 5.2.1 and 5.2.2,*

$$\mathbf{C} = [\mathbf{0} \mid -1], \quad \Phi(s) = \left[ \begin{array}{c|c} \widehat{\Phi}(s) & \mathbf{0} \\ \hline \widehat{\mathbf{C}} & 1 \end{array} \right], \quad \mathbf{B} = \left[ \begin{array}{c} \widehat{\mathbf{B}} \\ 0 \end{array} \right], \quad (5.13)$$

is a  $R$ -controllable and  $R$ -observable realization of size  $n + 2$ .

### 5.2.2 Two variable realizations

We assume the rational function given as

$$\mathbf{g}(s, t) = \frac{\sum_{i=0}^n \mathbf{b}_i(t) s^i}{\sum_{i=0}^n \mathbf{a}_i(t) s^i}, \quad \text{with } \mathbf{a}_i(t) = \sum_{j=0}^m \alpha_{i,j} t^j, \quad \mathbf{b}_i(t) = \sum_{j=0}^m \beta_{i,j} t^j.$$

We also use the following notation

$$\mathbf{A} := \begin{bmatrix} \alpha_{00} & \alpha_{10} & \cdots & \alpha_{n0} \\ \alpha_{01} & \alpha_{11} & \cdots & \alpha_{n1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{0m} & \alpha_{1m} & \cdots & \alpha_{nm} \end{bmatrix}, \quad \mathbb{B} := \begin{bmatrix} \beta_{00} & \beta_{10} & \cdots & \beta_{n0} \\ \beta_{01} & \beta_{11} & \cdots & \beta_{n1} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{0m} & \beta_{1m} & \cdots & \beta_{nm} \end{bmatrix}. \quad (5.14)$$

**Theorem 5.2.1.** *The following descriptor realization of the two-variable rational function  $\mathbf{g}$  is of size  $n + 2m + 2$ :*

$$\mathbf{C} = [\mathbf{0} \mid \mathbf{0} \mid \mathbf{e}_{m+1}^T], \quad \mathbf{B}^T = \left[ \begin{array}{c|c|c} \mathbf{0} & \mathbf{e}_1 & \mathbf{0} \end{array} \right], \quad (5.15)$$

$$\Phi(s, t) = \left[ \begin{array}{c|c|c} \mathbf{J}(s; n) & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{A} & \mathbf{J}^*(t; m) & \mathbf{0} \\ \hline \mathbb{B} & \mathbf{0} & [\mathbf{J}^*(t; m) \ \mathbf{e}_1] \end{array} \right], \quad (5.16)$$

with  $\Phi$  partitioned as:  $(n, m+1, m+1) \times (n+1, m, m+1)$ . It is  $R$ -controllable and  $R$ -observable, i.e.  $[\Phi(s, t), \mathbf{B}]$  and  $[\Phi^*(s, t), \mathbf{C}^*]$  are full rank,  $\forall s, t \in \mathbb{C}$ .

For  $\mathbf{g}$  expressed in the Lagrange basis, we need  $\mathbf{A}$  and  $\mathbb{B}$  as in Eq. (5.14), where

the quantities  $\alpha_{i,j}$  are obtained from the right null space of the two variable Loewner matrix  $\mathbb{L}$  and  $\beta_{i,j} = \alpha_{i,j} \mathbf{w}_{i,j}$ . We also need a vector  $\mathbf{p} \in \mathbb{C}^{m+1}$  such that  $[\mathbf{J}_{\text{lag}}^*(t; \pi_j, m), \mathbf{p}]$  is a unimodular matrix in  $t$ . The entries of this vector can be chosen as:

$$p_i = \frac{1}{\prod_{j=0, j \neq i}^m (\pi_i - \pi_j)}, \quad (5.17)$$

in which case  $\det [\mathbf{J}_{\text{lag}}^*(t; \pi_j, m), \mathbf{p}] = (-1)^m$ .

**Theorem 5.2.2.** *This is a descriptor realization of the two-variable rational function  $\mathbf{g}$  expressed in Lagrange form (5.3)*

$$\mathbf{C} = [ \mathbf{0} \mid \mathbf{0} \mid \mathbf{e}_{m+1}^T ], \quad \mathbf{B} = \left[ \begin{array}{c|c|c} \mathbf{0} & \mathbf{p} & \mathbf{0} \end{array} \right] \quad (5.18)$$

$$\Phi(s, t) = \left[ \begin{array}{c|c|c} \mathbf{J}_{\text{lag}}(s; \lambda_i, n) & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{A} & \mathbf{J}_{\text{lag}}^*(t; \pi_j, m) & \mathbf{0} \\ \hline \mathbf{B} & \mathbf{0} & [\mathbf{J}_{\text{lag}}^*(t; \pi_j, m), \mathbf{p}] \end{array} \right], \quad (5.19)$$

of dimension  $n + 2m + 2$ ,  $R$ -controllable and  $R$ -observable.

### 5.3 Obtaining a real realization

This section discusses the issue of obtaining a real realization from the one presented in Theorem 5.2.2 when one of the variables, for example  $s$ , the frequency, is measured on the imaginary axis, and the other one,  $t$ , the parameter, leads to real measurements.

We first discuss this for an example, which we later generalize.

**Example 5.3.1.** *Consider a system described by the transfer function  $\mathbf{H}(s, t) = \frac{1+2s+3t+4st}{5+6s+7t+8st}$ . We take measurements at  $s = \pm j$  and  $t = 1, t = 2$ , so the Lagrange basis is  $(s-j)(t+1), (s-j)(t+2), (s+j)(t+1), (s+j)(t+2)$ . The order in  $s$  is  $n = 1$ , and in  $t$*

is  $m = 1$ . Therefore,

$$\mathbb{A} = \begin{bmatrix} -2 + 38j & -38 - 2j \\ 19 - 180j & 181 \end{bmatrix}, \quad \mathbb{B} = \begin{bmatrix} -2 + 38j & -38 - 2j \\ 15 - 104j & 105 - 4j \end{bmatrix}$$

and the resulting realization, following Theorem 5.2.2 is

$$\mathbf{g}(s, t) = \left[ \begin{array}{ccc|cc} & s-j & -s-j & 0 & 0 & 0 \\ & -2+38j & -38-2j & t+1 & 0 & 0 \\ 0 & 0 & 0 & -t-2 & 0 & 0 \\ & -2+38j & -38-2j & 0 & t+1 & -1 \\ & 15-104j & 105-4j & 0 & -t-2 & 1 \end{array} \right]^{-1} \begin{bmatrix} 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

This realization contains complex quantities in the  $(1, 1)$ -block, namely  $\mathbf{J}_{lag}(s; \lambda_i, n)$ , as well as in  $\mathbb{A}$  and  $\mathbb{B}$ . To obtain a real realization, one may think of scaling the first two columns of  $\Phi(s, t)$ , by  $k_1$  and  $k_2$ . Unfortunately, no such scaling exists, but for  $k = \text{conj}(\mathbb{A}_{11} + \mathbb{A}_{12}) = -40 - 36j$ , the scaled  $\mathbb{A}$  and  $\mathbb{B}$  are

$$\mathbb{A}_s = \mathbb{A} \frac{k}{181 \cdot 2 \cdot 4} = \begin{bmatrix} 1-j & 1+j \\ -5+9/2j & -5-9/2j \end{bmatrix}, \quad \mathbb{B}_s = \mathbb{B} \frac{k}{181 \cdot 2 \cdot 4} = \begin{bmatrix} 1-j & 1+j \\ -3+5/2j & -3-5/2j \end{bmatrix}.$$

Next, we apply the transformation  $\mathbf{M} = \text{blkdiag}(\Pi, \mathbf{I}_m, \mathbf{I}_{m+1})$  to  $\Phi$  on the right where

$$\Pi = \begin{bmatrix} 1 & j \\ 1 & -j \end{bmatrix}.$$

The first  $n + 1 = 2$  columns of new  $\Phi$  are

$$\left[ \mathbf{J}_{lag}(s; \lambda_i, n)\Pi \quad \mathbb{A}_s\Pi \quad \mathbb{B}_s\Pi \right] = \begin{bmatrix} -2j & 2 & -10 & 2 & -6 \\ 2js & 2 & -9 & 2 & -5 \end{bmatrix}^T,$$

while  $\mathbf{C}$  stays unchanged by multiplication with  $\mathbf{M}$  on right. Next, multiply the new

$\Phi$  by  $\mathbf{N} = \text{blkdiag}(\mathbf{N}_1, \mathbf{I}_{m+1}, \mathbf{I}_{m+1})$  on left, where  $\mathbf{N}_1 = \begin{bmatrix} j \\ 2 \end{bmatrix}$  to make  $\Phi$  completely real ( $\mathbf{B}$  is not affected by pre-multiplication with  $\mathbf{N}$ ). The realization containing only real quantities is

$$\mathbf{g}(s, t) = \left[ \begin{array}{cc|c|c|c} 0 & 0 & 0 & 0 & -1 \end{array} \right] \begin{bmatrix} 1 & -s & 0 & 0 & 0 \\ 2 & 2 & t+1 & 0 & 0 \\ -10 & -9 & -t-2 & 0 & 0 \\ 2 & 2 & 0 & t+1 & -1 \\ -6 & -5 & 0 & -t-2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

In the general case, assume that the measurements in  $s$  come in complex conjugate pairs:  $\lambda_1 = j\omega_1, \lambda_2 = \bar{\lambda}_1 = -j\omega_1, \dots, \lambda_n = j\omega_{\frac{n+1}{2}}, \lambda_{n+1} = \bar{\lambda}_n = -j\omega_{\frac{n+1}{2}}$ . Construct the realization as in Theorem 5.2.2 and, afterwards, apply the following steps:

1. Scale  $\mathbf{A}$  and  $\mathbf{B}$  by  $k$ , where  $k = \text{conj}(\mathbf{A}_{11} + \mathbf{A}_{12})$ .
2. Post-multiply  $\Phi$  by  $\mathbf{M} = \text{blkdiag}(\mathbf{\Pi}, \mathbf{I}_m, \mathbf{I}_{m+1})$ , where  $\mathbf{\Pi} = \text{blkdiag}(\mathbf{\Pi}_1, \dots, \mathbf{\Pi}_1)$ ,

$$\mathbf{\Pi} \in \mathbb{C}^{(n+1) \times (n+1)} \text{ and } \mathbf{\Pi}_1 = \begin{bmatrix} 1 & j \\ 1 & -j \end{bmatrix}.$$

3. Pre-multiply the new  $\Phi$  by  $\mathbf{N} = \text{blkdiag}(\mathbf{N}_1, \mathbf{I}_{m+1}, \mathbf{I}_{m+1})$ , where

$$\mathbf{N}_1 = \frac{1}{2} \begin{bmatrix} j & & & & & \\ -1 & 1 & 1 & & & \\ 0 & j & -j & & & \\ & & & \ddots & & \\ -1 & & & & 1 & 1 \\ 0 & & & & j & -j \end{bmatrix} \in \mathbb{C}^{n \times n} \Rightarrow \Phi_{11} = \begin{bmatrix} \omega_1 & -s & & & & \\ s & \omega_1 & -s & -\omega_2 & & \\ 0 & 0 & -\omega_2 & s & & \\ & & & & \ddots & \\ s & \omega_1 & & & & -s & -\omega_{\frac{n+1}{2}} \\ 0 & 0 & & & & -\omega_{\frac{n+1}{2}} & s \end{bmatrix}.$$

This way, we obtain the following realization:



$$\mathbf{C} = [ \mathbf{0} \mid \mathbf{0} \mid \mathbf{e}_{m+1}^T ], \quad \mathbf{B} = \left[ \begin{array}{c|c|c} \mathbf{0} & \mathbf{p} & \mathbf{0} \end{array} \right], \quad (5.20)$$

$$\Phi(s, t) = \left[ \begin{array}{c|c|c} \Phi_{11} = \mathbf{N}_1 \mathbf{J}_{\text{lag}}(s; \lambda_i, n) \Pi & \mathbf{0} & \mathbf{0} \\ \hline k \Delta \Pi & \mathbf{J}_{\text{lag}}^*(t; \pi_j, m) & \mathbf{0} \\ \hline k \mathbb{B} \Pi & \mathbf{0} & [\mathbf{J}_{\text{lag}}^*(t; \pi_j, m), \mathbf{p}] \end{array} \right], \quad (5.21)$$

with  $k = \text{conj}(\mathbb{A}_{11} + \mathbb{A}_{12})$ , which contains only real quantities.

**Example 5.3.2.** We consider a more involved example. The transfer function is of order  $n = 3$  in  $s$  and order  $m = 1$  in  $t$ :  $\mathbf{H}(s, t) = \frac{3s^2t + st + 2}{s^3t + s^2t + t - 1}$ , and the measurements we consider are:  $s_i = \{j, -j, 2j, -2j, 3j, -3j, 4j, -4j\}$  and  $t_j = \{3, 5, 7, 9\}$ .

The usual realization is

$$\Phi(s, t) = \left[ \begin{array}{cccc|ccc} s - i & -i - s & 0 & 0 & 0 & 0 & 0 \\ s - i & 0 & 2i - s & 0 & 0 & 0 & 0 \\ s - i & 0 & 0 & -2i - s & 0 & 0 & 0 \\ \hline \frac{13}{116} - \frac{11}{116}i & \frac{17}{116} - \frac{1}{116}i & -\frac{25}{58} + \frac{49}{116}i & -\frac{35}{58} - \frac{1}{116}i & t - 3 & 0 & 0 \\ -\frac{23}{116} + \frac{15}{116}i & -\frac{27}{116} + \frac{5}{116}i & \frac{21}{29} - \frac{20}{29}i & 1 & 5 - t & 0 & 0 \\ \hline -\frac{1}{4} - \frac{1}{4}i & -\frac{1}{116} + \frac{41}{116}i & \frac{49}{116} + \frac{79}{116}i & -\frac{19}{116} - \frac{91}{116}i & 0 & t - 3 & -1/2 \\ \frac{51}{116} + \frac{55}{116}i & -\frac{1}{116} - \frac{75}{116}i & -\frac{83}{116} - \frac{135}{116}i & \frac{33}{116} + \frac{155}{116}i & 0 & 5 - t & 1/2 \end{array} \right]$$

$$\mathbf{C} = \left[ \begin{array}{cccc|c|c} 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right]$$

$$\mathbf{B} = \left[ \begin{array}{ccc|cc|cc} 0 & -1/2 & 1/2 & 0 & 0 & 0 & 0 \end{array} \right]^T, \quad (5.22)$$

while the real realization is

$$\Phi(s, t) = \left[ \begin{array}{cccc|ccc} 1 & -s & 0 & 0 & 0 & 0 & 0 \\ s & 1 & -s & -2 & 0 & 0 & 0 \\ 0 & 0 & -2 & s & 0 & 0 & 0 \\ \frac{9}{116} & \frac{3}{116} & -\frac{9}{29} & -\frac{15}{116} & t-3 & 0 & 0 \\ -\frac{15}{116} & -\frac{3}{116} & \frac{15}{29} & \frac{6}{29} & 5-t & 0 & 0 \\ -\frac{9}{116} & \frac{21}{116} & \frac{9}{116} & -\frac{51}{116} & 0 & t-3 & -1/2 \\ \frac{15}{116} & -\frac{39}{116} & -\frac{15}{116} & \frac{3}{4} & 0 & 5-t & 1/2 \end{array} \right] \quad (5.23)$$

$$\mathbf{C} = \left[ \begin{array}{cccc|c|cc} 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \quad (5.24)$$

$$\mathbf{B} = \left[ \begin{array}{ccc|cc|cc} 0 & -1/2 & 1/2 & 0 & 0 & 0 & 0 \end{array} \right]^T. \quad (5.25)$$

## 5.4 Generalized state-space realizations for MIMO systems

The goal is to generalize the realizations presented in Sect. 5.2 to MIMO systems. In the following, we consider matrix transfer functions, in other words, systems with more than one input and output, rather than scalar (SISO) transfer functions.

### 5.4.1 One variable case

We assume the same number of input and output ports,  $p$ . We use the MIMO Loewner matrix  $\mathbb{L}^{(M)}$  built from matrix data  $\mathbb{L}_{i,j}^{(M)} = \frac{\mathbf{V}_i - \mathbf{W}_j}{\mu_i - \lambda_j}$  (instead of Eq.(4.8)) where  $\mathbf{V}_i, \mathbf{W}_j \in \mathbb{C}^{p \times p}$  and  $\mathbf{H}(\lambda_j) = \mathbf{W}_j$ , as well as  $\mathbf{H}(\mu_i) = \mathbf{V}_i$ ,  $i, j = 0, \dots, k$  [75]. Consequently, there are  $p$  vectors in the null space:  $\mathbf{a} = \begin{bmatrix} \alpha_0 & \alpha_1 & \dots & \alpha_k \end{bmatrix} \in \mathbb{R}^{p \times (k+1)p}$ , where  $\alpha_i \in \mathbb{C}^{p \times p}$ ,  $i = 0, \dots, k$  and  $\mathbf{b} = \begin{bmatrix} \beta_0 & \beta_1 & \dots & \beta_k \end{bmatrix} \in \mathbb{R}^{p \times (k+1)p}$ , with  $\beta_i = \mathbf{W}_i \alpha_i$ ,  $i = 0, \dots, k$ . We define the MIMO analogue of Eq. (5.11):

$$\mathbf{J}(s; \lambda_i, k) = \begin{bmatrix} (s-\lambda_0)\mathbf{I}_p & (\lambda_1-s)\mathbf{I}_p & & & \\ (s-\lambda_0)\mathbf{I}_p & & (\lambda_2-s)\mathbf{I}_p & & \\ \vdots & & & \ddots & \\ (s-\lambda_0)\mathbf{I}_p & & & & (\lambda_k-s)\mathbf{I}_p \end{bmatrix} \in \mathbb{R}^{kp \times (k+1)p}[s]. \quad (5.26)$$

**Theorem 5.4.1.** *The following is a descriptor realization of  $\mathbf{H}$  of dimension  $(k+1)p$ :*

$$\widehat{\mathbf{C}} = \mathbf{b}, \quad \widehat{\Phi}(s) = \begin{bmatrix} \mathbf{J}(s; \lambda_i, k) \\ \mathbf{a} \end{bmatrix}, \quad \widehat{\mathbf{B}} = \begin{bmatrix} \mathbf{0}_{kp \times p} \\ \mathbf{I}_p \end{bmatrix}. \quad (5.27)$$

*Proof.* Interpolation conditions  $\mathbf{H}(\lambda_i) = \widehat{\mathbf{C}}\widehat{\Phi}^{-1}(\lambda_i)\widehat{\mathbf{B}} = \mathbf{W}_i$  and  $\mathbf{H}(\mu_i) = \widehat{\mathbf{C}}\widehat{\Phi}^{-1}(\mu_i)\widehat{\mathbf{B}} = \mathbf{V}_i$  should be satisfied. For the first, we denote  $\Phi^{-1}(\lambda_i)\widehat{\mathbf{B}}$  by  $\mathbf{x}$ , so  $\Phi(\lambda_i)\mathbf{x} = \widehat{\mathbf{B}}$ :

$$\begin{bmatrix} (\lambda_i - \lambda_1)\mathbf{I}_p & (\lambda_2 - \lambda_i)\mathbf{I}_p & & & \\ \vdots & & \ddots & & \\ (\lambda_i - \lambda_1)\mathbf{I}_p & & & \underbrace{(\lambda_i - \lambda_i)\mathbf{I}_p}_{\mathbf{0}_p} & \\ \vdots & & & & \ddots \\ (\lambda_i - \lambda_1)\mathbf{I}_p & & & & (\lambda_{k+1} - \lambda_i)\mathbf{I}_p \\ \alpha_0 & \alpha_1 & \dots & \alpha_i & \dots & \alpha_k \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{i-1} \\ \vdots \\ \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_p \\ \vdots \\ \mathbf{0}_p \\ \vdots \\ \mathbf{0}_p \\ \mathbf{I}_p \end{bmatrix}$$

$$\Rightarrow \begin{cases} (\lambda_i - \lambda_1)\mathbf{x}_1 = (\lambda_i - \lambda_2)\mathbf{x}_2 \\ \vdots \\ (\lambda_i - \lambda_1)\mathbf{x}_1 = \mathbf{0} \\ \vdots \\ \alpha_0\mathbf{x}_1 + \dots + \alpha_{i-1}\mathbf{x}_i + \dots + \alpha_k\mathbf{x}_{k+1} = \mathbf{I}_p \end{cases} \Rightarrow \begin{cases} \mathbf{x}_1 = \mathbf{0} \\ \vdots \\ \mathbf{x}_{k+1} = \mathbf{0} \\ \alpha_{i-1}\mathbf{x}_i = \mathbf{I}_p \Rightarrow \mathbf{x}_i = \alpha_{i-1}^{-1} \end{cases}$$

for  $i \neq 1$ , where each  $\mathbf{x}_i$  is of dimension  $p \times p$  and  $\alpha_{i-1}$  invertible. Similarly, for  $i = 1$ ,  $\mathbf{x}_1 = \alpha_0^{-1}$ , so  $\mathbf{x}_i = \alpha_{i-1}^{-1}, \forall i = 1, \dots, k+1$ . Therefore,

$$\mathbf{H}(\lambda_i) = \widehat{\mathbf{C}}\widehat{\Phi}^{-1}(\lambda_i)\widehat{\mathbf{B}} = \widehat{\mathbf{C}}\mathbf{x} = [\mathbf{W}_1\alpha_0 \quad \mathbf{W}_2\alpha_1 \quad \dots \quad \mathbf{W}_{k+1}\alpha_k] \begin{bmatrix} \mathbf{0} \\ \vdots \\ \alpha_{i-1}^{-1} \\ \dots \\ \mathbf{0} \end{bmatrix} = \mathbf{W}_i.$$

This part of the proof holds for any invertible  $\alpha_i$ .

For the second condition, let us denote  $\Phi^{-1}(\mu_i)\widehat{\mathbf{B}}$  as  $\mathbf{y}$ , so  $\Phi(\mu_i)\mathbf{y} = \widehat{\mathbf{B}}$ :

$$\begin{bmatrix} (\mu_i - \lambda_1)\mathbf{I}_p & (\lambda_2 - \mu_i)\mathbf{I}_p & \dots & \\ \vdots & \vdots & \ddots & \\ (\mu_i - \lambda_1)\mathbf{I}_p & \dots & & (\lambda_{k+1} - \mu_i)\mathbf{I}_p \\ \alpha_0 & \alpha_1 & \dots & \alpha_k \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{0}_p \\ \mathbf{0}_p \\ \vdots \\ \mathbf{I}_p \end{bmatrix} \quad (5.28)$$

$$\Rightarrow \begin{cases} \mathbf{y}_1 = \frac{\mu_i - \lambda_1}{\mu_i - \lambda_2} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_k = \frac{\mu_i - \lambda_1}{\mu_i - \lambda_{k+1}} \mathbf{y}_0 \\ \alpha_0 \mathbf{y}_0 + \dots + \alpha_k \mathbf{y}_k = \mathbf{I}_p \Rightarrow \alpha_0 \mathbf{y}_0 + \alpha_1 \frac{\mu_i - \lambda_1}{\mu_i - \lambda_2} \mathbf{y}_0 + \dots + \alpha_k \frac{\mu_i - \lambda_1}{\mu_i - \lambda_{k+1}} \mathbf{y}_0 = \mathbf{I}_p. \end{cases} \quad (5.29)$$

Since the column vector  $\begin{bmatrix} \alpha_0 & \dots & \alpha_k \end{bmatrix}^T$  is in the nullspace of the MIMO Loewner matrix  $\mathbb{L}^{(M)}$ , by multiplying each row of  $\mathbb{L}^{(M)}$  with this vector, we get  $\mathbf{0}$ . Therefore,

$$\frac{\mathbf{V}_i - \mathbf{W}_1}{\mu_i - \lambda_1} \alpha_0 + \frac{\mathbf{V}_i - \mathbf{W}_2}{\mu_i - \lambda_2} \alpha_1 \dots + \frac{\mathbf{V}_i - \mathbf{W}_{k+1}}{\mu_i - \lambda_{k+1}} \alpha_k = \mathbf{0} \mid \cdot (\mu_i - \lambda_1) \mathbf{y}_0 \Rightarrow \quad (5.30)$$

$$\mathbf{V}_i \left( \underbrace{\alpha_0 \mathbf{y}_0 + \alpha_1 \frac{\mu_i - \lambda_1}{\mu_i - \lambda_2} \mathbf{y}_0 + \dots + \alpha_k \frac{\mu_i - \lambda_1}{\mu_i - \lambda_{k+1}} \mathbf{y}_0}_{\mathbf{I}_p \text{ from Eq.(5.29)}} \right) \quad (5.31)$$

$$= \mathbf{W}_1 \alpha_0 \mathbf{y}_0 + \mathbf{W}_2 \alpha_1 \frac{\mu_i - \lambda_1}{\mu_i - \lambda_2} \mathbf{y}_0 + \dots + \mathbf{W}_{k+1} \alpha_k \frac{\mu_i - \lambda_1}{\mu_i - \lambda_{k+1}} \mathbf{y}_0.$$

After performing obvious simplifications, we arrive at

$$\begin{aligned}
 \mathbf{V}_i &= \mathbf{W}_1 \alpha_0 y_0 + \mathbf{W}_2 \alpha_1 \underbrace{\frac{\mu_i - \lambda_1}{\mu_i - \lambda_2} y_0}_{y_1} + \dots + \mathbf{W}_{k+1} \alpha_k \underbrace{\frac{\mu_i - \lambda_1}{\mu_i - \lambda_{k+1}} y_0}_{y_k} \\
 \mathbf{V}_i &= \underbrace{\begin{bmatrix} \mathbf{W}_1 \alpha_0 & \mathbf{W}_2 \alpha_1 & \dots & \mathbf{W}_{k+1} \alpha_k \end{bmatrix}}_{\widehat{\mathbf{C}}} \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \dots \\ y_k \end{bmatrix}}_{\Phi^{-1}(\mu_i) \widehat{\mathbf{B}}},
 \end{aligned}$$

so, indeed,  $\mathbf{H}(\mu_i) = \widehat{\mathbf{C}} \widehat{\Phi}^{-1}(\mu_i) \widehat{\mathbf{B}} = \mathbf{V}_i$ . This proof also holds for Lemma 5.2.2 which is a special case of Theorem 5.4.1 for one input and one output ( $p = 1$ ).  $\square$

**Remark 5.4.1.** *The dimension of this realization should be close to the order of the minimum realization with  $\mathbf{D} = \mathbf{0}$ , which is  $n + p$ , with  $n$ , the McMillan degree, and  $p$ , the number of ports (for the general case of an invertible non-zero  $\mathbf{D}$ -term). We choose  $k$  such that the size of our realization is as close as possible to  $n + p$ :  $n + p < (k + 1)p$ . Thus,  $k$  is  $\lceil \frac{n}{p} \rceil$ .*

**Example 5.4.1.** *Consider the transfer function*

$$\mathbf{H}(s) = \begin{bmatrix} \frac{2+5s}{1+2s} & \frac{3+7s}{1+2s} \\ \frac{1+6s}{1+2s} & \frac{4+9s}{1+2s} \end{bmatrix}$$

*and samples at  $\lambda_0=1$ ,  $\lambda_1=2$ ,  $\mu_0=3$ ,  $\mu_1=4$ . The MIMO Loewner matrix is*

$$\mathbb{L}^{(M)} = \begin{bmatrix} 1/21 & 1/21 & 1/35 & 1/35 \\ 4/21 & 1/21 & 4/35 & 1/35 \\ 1/27 & 1/27 & 1/45 & 1/45 \\ 4/27 & 1/27 & 4/45 & 1/45 \end{bmatrix}.$$

*It has dimension 4 and its nullspace is of dimension 2:*

$$\mathbf{c} = \left\{ \left[ \begin{array}{c} -3/5 \\ 0 \\ 1 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ -3/5 \\ 0 \\ 1 \end{array} \right] \right\},$$

thus,  $\alpha_0 = -3/5\mathbf{I}_2$  and  $\alpha_1 = \mathbf{I}_2$ . In fact,  $\alpha_i$  can be chosen as any matrices such that  $[\alpha_0 \ \alpha_1]^T$  is in the right nullspace of  $\mathbb{L}^{(M)}$ . In this case, the McMillan degree  $n = 2$ , so  $k = 1$ , and the minimal realization with  $\mathbf{D} = \mathbf{0}$  is of order  $n + p = 2 + 2 = 4$ . This is the same as the dimension of the realization we can construct using Theorem 5.4.1:

$$\mathbf{C} = \begin{bmatrix} -7/5 & -2 & 12/5 & 17/5 \\ -7/5 & -13/5 & 13/5 & 22/5 \end{bmatrix},$$

$$\Phi(s) = \begin{bmatrix} (s-1)\mathbf{I}_2 & (2-s)\mathbf{I}_2 \\ -3/5\mathbf{I}_2 & \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{I}_2 \end{bmatrix},$$

recovers the original MIMO transfer function.

### 5.4.2 Two variable case

Assuming the same number of inputs and outputs,  $p$ , we are now using the 2D MIMO Loewner matrix  $\mathbb{L}^{(M)}$  built using MIMO data  $\ell_{i,j}^{k,l} = \frac{\mathbf{V}_{k,l} - \mathbf{W}_{i,j}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)}$  (instead of  $\ell_{i,j}^{k,l} = \frac{v_{k,l} - w_{i,j}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)}$  in the scalar case) where  $\mathbf{V}_{k,l}, \mathbf{W}_{i,j} \in \mathbb{C}^{p \times p}$  and  $\mathbf{H}(\lambda_i, \pi_j) = \mathbf{W}_{i,j}$ , as well as  $\mathbf{H}(\mu_k, \nu_l) = \mathbf{V}_{k,l}$ ,  $i, k = 0, \dots, k_1$ ,  $j, l = 0, \dots, k_2$ .

We need  $\mathbf{q} \in \mathbb{C}^{(k_2+1)p \times p}$  such that  $[\mathbf{J}^*(t; \pi_j, k_2), \mathbf{q}]$  is unimodular in  $t$ . We choose

$$q_i = \frac{1}{\prod_{j=0, j \neq i}^{k_2} (\pi_i - \pi_j)},$$

so  $\mathbf{q} = \begin{bmatrix} q_0 \cdot \mathbf{I}_p & \dots & q_{k_2} \cdot \mathbf{I}_p \end{bmatrix}^T$ , in which case  $\det[\mathbf{J}(t; \pi_j, k_2), \mathbf{q}] = (-1)^{k_2}$ .

The right null space of the two variable Loewner matrix  $\mathbb{L}^{(M)}$  from MIMO data is of dimension  $p$ , so  $\mathbb{L}^{MIMO} \begin{bmatrix} \alpha_{00} & \dots & \alpha_{0k_2} & \dots & \alpha_{k_10} & \dots & \alpha_{k_1k_2} \end{bmatrix}^T = \mathbf{0}$ , where each

$\alpha_{i,j}$  is of dimension  $p \times p$ . We arrange the entries from the nullspace into the matrix

$$\mathbf{A} := \begin{bmatrix} \alpha_{00} & \alpha_{10} & \cdots & \alpha_{k_1 0} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{0k_2} & \alpha_{1k_2} & \cdots & \alpha_{k_1 k_2} \end{bmatrix}, \quad (5.32)$$

and

$$\mathbf{B} := \begin{bmatrix} \beta_{00} & \beta_{10} & \cdots & \beta_{k_1 0} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{0k_2} & \beta_{1k_2} & \cdots & \beta_{k_1 k_2} \end{bmatrix}, \quad \text{where } \beta_{i,j}^{(1:p,1:p)} = \mathbf{W}_{i,j} \alpha_{i,j}. \quad (5.33)$$

**Theorem 5.4.2.** *The following is a descriptor realization of the MIMO two-variable rational function  $\mathbf{H}(s, t)$*

$$\mathbf{C} = -[\mathbf{0} \mid \mathbf{0} \mid \mathbf{e}_{k_2+1}^T \cdot \mathbf{I}_p], \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{q} & \mathbf{0} \end{bmatrix} \quad (5.34)$$

$$\Phi(s, t) = \begin{bmatrix} \mathbf{J}(s; \lambda_i, k_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{A} & \mathbf{J}^*(t; \pi_j, k_2) & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & [\mathbf{J}^*(t; \pi_j, k_2), \mathbf{q}] \end{bmatrix}, \quad (5.35)$$

of dimension  $(k_1 + 2k_2 + 2)p$ .

*Proof.* See Appendix A.1. □

**Remark 5.4.2.** *Similarly to the one variable case,  $k_1$  can be chosen as  $\left\lceil \frac{n}{p} \right\rceil$ , while  $k_2$  can be chosen as  $\left\lceil \frac{m}{p} \right\rceil$ .*

**Example 5.4.2.** *Consider the transfer function of order  $n = 2$  in  $s$  and  $m = 2$  in  $t$ ,*

$$\mathbf{H}(s) = \begin{bmatrix} \frac{s+5t+9st-1}{2s+3t+st-1} & \frac{2s+6t+10st-2}{2s+3t+st-1} \\ \frac{3s+7t+11st-3}{2s+3t+st-1} & \frac{4s+8t+12st-4}{2s+3t+st-1} \end{bmatrix},$$

as well as the measurement points  $\lambda_0 = 2$ ,  $\lambda_1 = 1/2$ ,  $\mu_0 = 3/2$ ,  $\mu_1 = 3$ ,  $\pi_0 = -1/2$ ,  $\pi_1 = -3/2$ ,  $\nu_0 = -1$ ,  $\nu_1 = -2$ . Note that, indeed, we have  $2 = k_1 + 1 = \left\lceil \frac{n}{p} \right\rceil + 1$  and similarly,  $2 = k_2 + 1 = \left\lceil \frac{m}{p} \right\rceil + 1$ . We build the 2D MIMO Loewner matrix of dimension  $8 = 2 \cdot 2 \cdot 2 = (k_1 + 1)(k_2 + 1)p$  with nullspace of dimension 2:

$$\mathbf{c} = \left\{ \begin{bmatrix} -2/21 & 0 & -6/7 & 0 & -1/3 & 0 & 1 & 0 \end{bmatrix}^T, \right. \\ \left. \begin{bmatrix} 0 & -2/21 & 0 & -6/7 & 0 & -1/3 & 0 & 1 \end{bmatrix}^T \right\},$$

so, after a reasonable scaling,  $\alpha_{00} = -2\mathbf{I}_2$ ,  $\alpha_{01} = -18\mathbf{I}_2$ ,  $\alpha_{10} = -7\mathbf{I}_2$  and  $\alpha_{11} = 21\mathbf{I}_2$ .

The matrix  $\mathbb{B}$  is

$$\begin{bmatrix} \beta_{00} & \beta_{10} \\ \beta_{01} & \beta_{11} \end{bmatrix} = \begin{bmatrix} 42 & 44 & -21 & -26 \\ 46 & 48 & -31 & -36 \\ -134 & -148 & 59 & 70 \\ -162 & -176 & 81 & 92 \end{bmatrix}.$$

Using the previous formulas we have the realization

$$\mathbf{C} = \begin{bmatrix} \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & -\mathbf{I}_2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_2 & -\mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 \end{bmatrix}^T, \\ \Phi(s, t) = \begin{bmatrix} (s-2)\mathbf{I}_2 & (1/2-s)\mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \alpha_{00} & \alpha_{10} & (t+1/2)\mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \alpha_{10} & \alpha_{11} & (-t-3/2)\mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \beta_{00} & \beta_{10} & \mathbf{0}_2 & (t+1/2)\mathbf{I}_2 & -\mathbf{I}_2 \\ \beta_{01} & \beta_{11} & \mathbf{0}_2 & (-t-3/2)\mathbf{I}_2 & \mathbf{I}_2 \end{bmatrix},$$

which recovers the original MIMO transfer function.

## 5.5 Numerical examples

### 5.5.1 RLC Circuit

In the circuit below, all elements have unit value except for the inductor  $L_2$  which is taken as the parameter  $p$ .



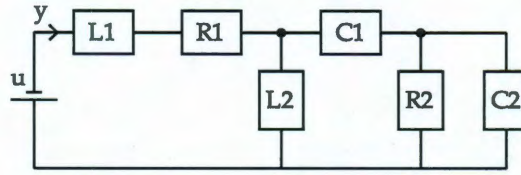


Figure 5.2: Electrical Circuit

Using the voltages across the capacitors and the currents through the inductors as state variables, the descriptor representation  $\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$ ,  $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$  is

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & -1 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{C} = \mathbf{B}^T.$$

The transfer function depends on two variables, namely the Laplace variable  $s$ , and the value of the inductor  $L_2$ , denoted by  $p$ :

$$\mathbf{g}(s, p) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} = \frac{ps^3 + ps^2 + 2s + 1}{s^4p + 2s^3p + 3s^2p + 2s^2 + sp + 3s + 1}.$$

We reconstruct this system using measurements at the following frequencies and values of the parameter:

$$\begin{bmatrix} \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \pi_1, \pi_2 \end{bmatrix} = \begin{bmatrix} 0, \frac{1}{2}, 1, \frac{3}{2}, 2, 0, -\frac{1}{2} \end{bmatrix}, \\ \begin{bmatrix} \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \nu_1, \nu_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{4}, -\frac{1}{2}, -1, -\frac{3}{2}, -2, 1, \frac{1}{2} \end{bmatrix}.$$

The resulting Loewner matrix is of dimension  $10 \times 10$ , has rank 9 and the following vector is in its nullspace  $\mathbf{c}^T = \begin{bmatrix} \frac{1}{8} & -\frac{1}{8} & -\frac{3}{2} & \frac{71}{64} & \frac{9}{2} & -\frac{15}{8} & -5 & -\frac{1}{64} & \frac{15}{8} & 1 \end{bmatrix}$ .

The realization given in Theorem 5.2.2 leads to

$$\Phi(s, p) = \left[ \begin{array}{ccccc|ccc} s & \frac{1}{2}-s & 0 & 0 & 0 & 0 & 0 & 0 \\ s & 0 & 1-s & 0 & 0 & 0 & 0 & 0 \\ s & 0 & 0 & \frac{3}{2}-s & 0 & 0 & 0 & 0 \\ s & 0 & 0 & 0 & 2-s & 0 & 0 & 0 \\ \hline \frac{1}{8} & -\frac{3}{2} & \frac{9}{2} & -5 & \frac{15}{8} & p & 0 & 0 \\ -\frac{1}{8} & \frac{71}{64} & -\frac{15}{8} & -\frac{1}{64} & 1 & -p-\frac{1}{2} & 0 & 0 \\ \hline \frac{1}{8} & -1 & \frac{9}{4} & -2 & \frac{5}{8} & 0 & p & -1 \\ -\frac{1}{8} & \frac{29}{32} & -\frac{3}{2} & \frac{19}{32} & \frac{1}{8} & 0 & -p-\frac{1}{2} & 1 \end{array} \right],$$

$$\mathbf{B}^T = \left[ \begin{array}{cccc|cc|cc} 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{array} \right],$$

$$\mathbf{C} = \left[ \begin{array}{cccc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right].$$

### 5.5.2 Measured S-parameters

We analyze an example from [76] consisting of two microstrip lines and a resistor-capacitor pair, which idealizes a 5cm interconnect link loaded by a device. The design parameter, the microstrip's width  $w$ , was considered for 15 values between 60 and 130  $\mu\text{m}$  in steps of 5  $\mu\text{m}$ . For each parameter value, the 2 port scattering matrix was computed for 100 frequencies between 10MHz and 10GHz. To avoid running into numerical difficulties, we scaled frequencies by  $10^{-6}$ . Also, to ensure a real system, we assumed the value of the transfer function evaluated at  $-j2\pi f_i$  to be the complex conjugate of the measurement provided at  $j2\pi f_i$ . Therefore, for each of the total 200 values for  $s$  and 15 values for  $w$ , we are provided with a  $2 \times 2$  matrix that represents the S-parameters measured for that particular frequency and width parameter. The goal is to construct a low order model that incorporates the dependence on both frequency and parameter and matches all measurements to a reasonable extent. We consider only the  $S_{1,1}$  entry for now.

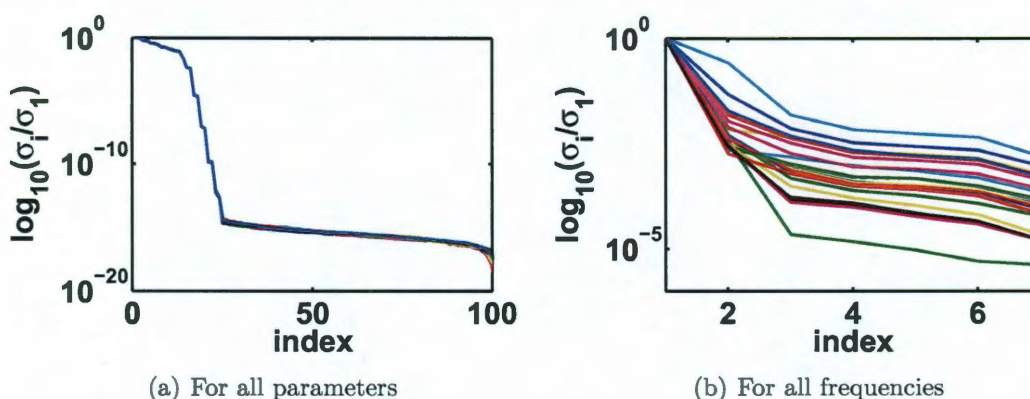


Figure 5.3: Singular value drop of the Loewner pencil

First, we compute the orders  $m$  and  $n$  of the two-variable rational model. When constructing the Loewner matrix pencil ([43, 44]) for each  $w$  and varying  $\omega_i = j2\pi f_i$ , we obtain the singular value drop in Fig. 5.3(a). Due to the noise in the data, the rank of the matrix, and consequently, the order  $n$ , is not obvious, but we decide to truncate at  $n = 17$ . Similarly, for the order  $m$ , we decide to use  $m = 3$  [73].

Fig. 5.4(a) shows the singular value drop of the two variable Loewner matrix of dimension  $(n + 1)(m + 1) = 18 \cdot 4 = 72$ . The last singular value is close to machine precision (it is not precisely 0 due to the noise in the data), so we consider the right null space to be the last right singular vector.

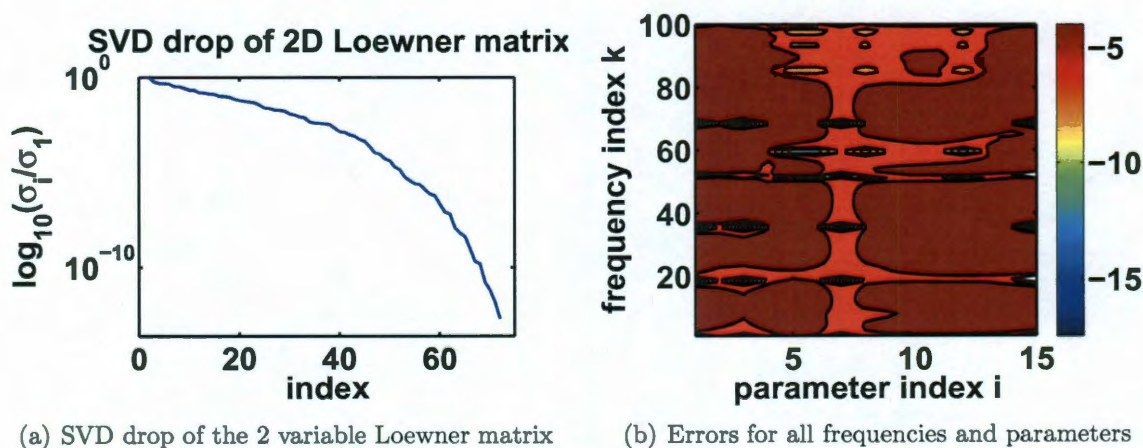


Figure 5.4: Plots for the  $S_{1,1}$  entry

Fig. 5.4(b) shows  $\log_{10}(\|\hat{S}^{(k,i)} - S^{(k,i)}\|)$ , namely the absolute errors on a logarithmic scale.

mic scale, with  $\hat{S}^{(k,i)}$ , the value of the rational model evaluated for frequency  $\omega_k$  and parameter  $w_i$  and  $S^{(k,i)}$ , the corresponding measurement. Errors are below  $5.9 \cdot 10^{-3}$ .

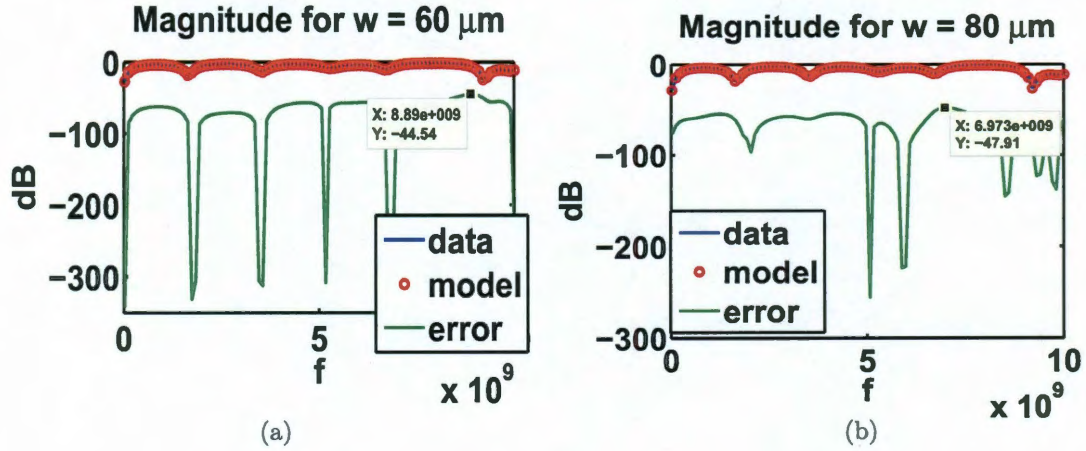


Figure 5.5: Magnitude plots

Fig. 5.5 shows the magnitude of  $S_{1,1}$  for parameters  $w_1 = 60 \mu\text{m}$  and  $w_5 = 80 \mu\text{m}$  which show large errors in Fig. 5.4(b). We plot the data, the model, as well as the absolute error on a dB scale. Indeed, the errors are below  $-40\text{dB}$ , as expected.

By employing the theory developed in Sect. 5.4.2, we are ready to consider the two-port matrix measurements of the S-parameters [75]. The computation consists of the following steps:

- Using all measurements in the real alternative approach (Sect. 4.2.3), we construct the Loewner matrix for each  $w$  and varying  $\omega_i = j2\pi f_i$ , to determine the order  $n$ . We compute the singular values of each Loewner matrix (shown in Fig. 5.6(a)). Due to the noise in the data, the rank of the matrix, and consequently, the degree  $n$ , is not obvious, but we decide to truncate at  $24 = n + p$ , where  $p = 2$ , so  $n = 22$  and  $k_1 = \left\lceil \frac{n}{p} \right\rceil = 11$ .
- After deciding to build models of order  $n + p$  with respect to the frequency  $s$ , we need to select the  $k_1 + 1$  frequency samples together with their complex conjugates adaptively (see Sect. 4.2.5), for each parameter value. At this point,

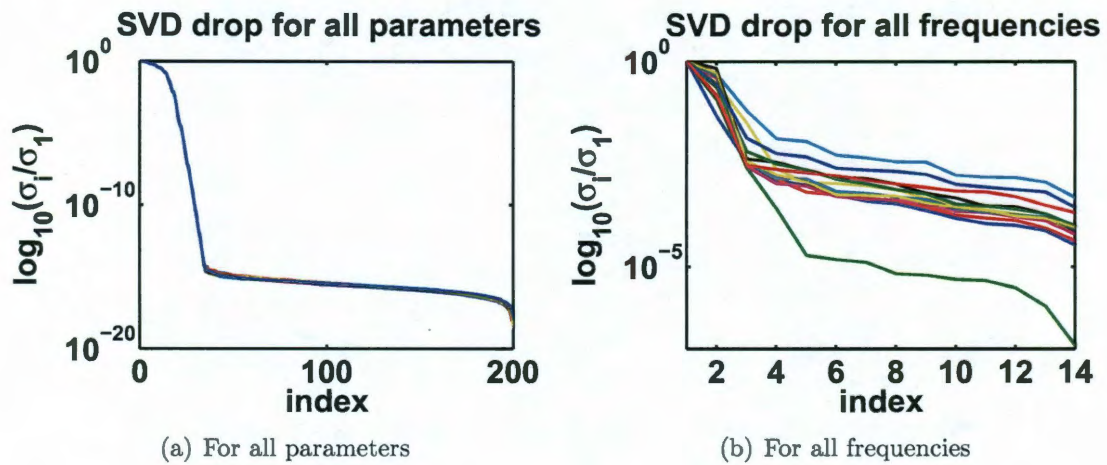


Figure 5.6: Singular value drop of the Loewner pencil

all systems constructed are stable as seen from Fig. 5.7 and the highest modeling errors are in the range  $10^{-4}$  for all parameter values.

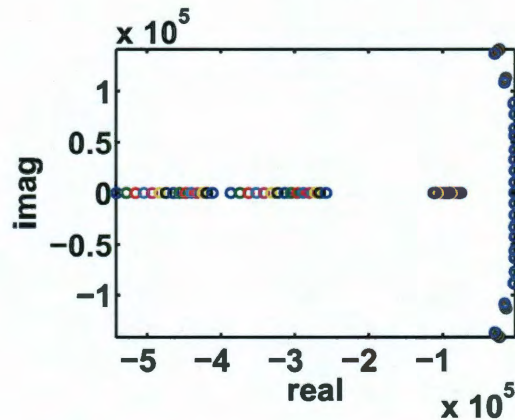


Figure 5.7: Poles after modeling

- We decide on the truncation order  $8 = m + p$ , where  $p = 2$  (so  $m = 6$  and  $k_2 = \left\lceil \frac{m}{p} \right\rceil = 3$ ) after computing the Loewner matrix from all parameter values, for each frequency (see the singular value drop of these various matrices in Fig. 5.6(b)). The resulting errors are in the range  $10^{-4}$ . Next, we select the  $k_2 + 1$  parameter samples adaptively for each frequency and, in the end, use the ones that appear most often.

- Fig. 5.8 shows the singular value drop of the 2D MIMO Loewner matrix of dimension  $(k_1 + 1)(k_2 + 1)p = 12 \cdot 4 \cdot 2 = 96$ . The last singular values are small (not precisely 0 due to the noise in the data), so we consider the null space as the last 2 right singular vectors.

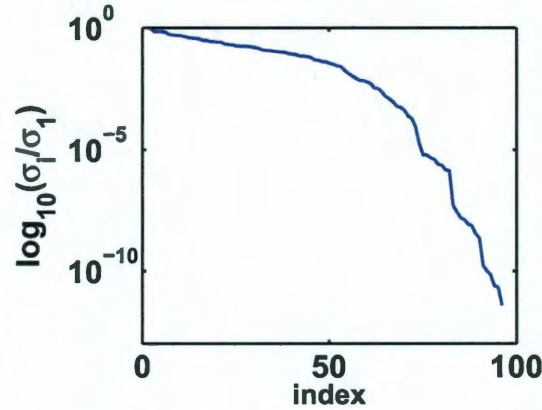


Figure 5.8: SVD drop of the two variable MIMO Loewner matrix

- We construct the realization following Sect. 5.4.2.

To check the accuracy, we computed the errors for all parameter and frequency values (displayed in Fig. 5.9). Fig. 5.9(a) shows the absolute value of the errors on a logarithmic scale, namely  $\log_{10}(\|\hat{S}_{1,1}^{(k,i)} - S_{1,1}^{(k,i)}\|)$ , where  $\hat{S}_{1,1}^{(k,i)}$  is the value of the  $S_{1,1}$  entry of the rational model evaluated for frequency  $j2\pi f_k$  (shown on the  $y$ -axis) and parameter  $w_i$  (shown on the  $x$ -axis), while  $S_{1,1}^{(k,i)}$  is the corresponding measured  $S_{1,1}$ . Similarly, the errors in the other entries are shown in Fig. 5.9(b), 5.9(c) and 5.9(d). The color bar shown on the right of each figure suggests that in the areas with dark red, the errors are in the  $10^{-4}$  range, while in those with orange, the errors are in the  $10^{-6}$  range. Nevertheless, all errors are below 50dB (the largest error is in fact  $2.4e - 3$ , while the largest relative error is  $9.3e - 3$ ).

**Remark 5.5.1.** *The dimension of the model is  $(k_1 + 2k_2 + 2)p = 2(11 + 2 \cdot 3 + 2) = 38$ , and it incorporates information about the frequency, as well as the width parameter.*

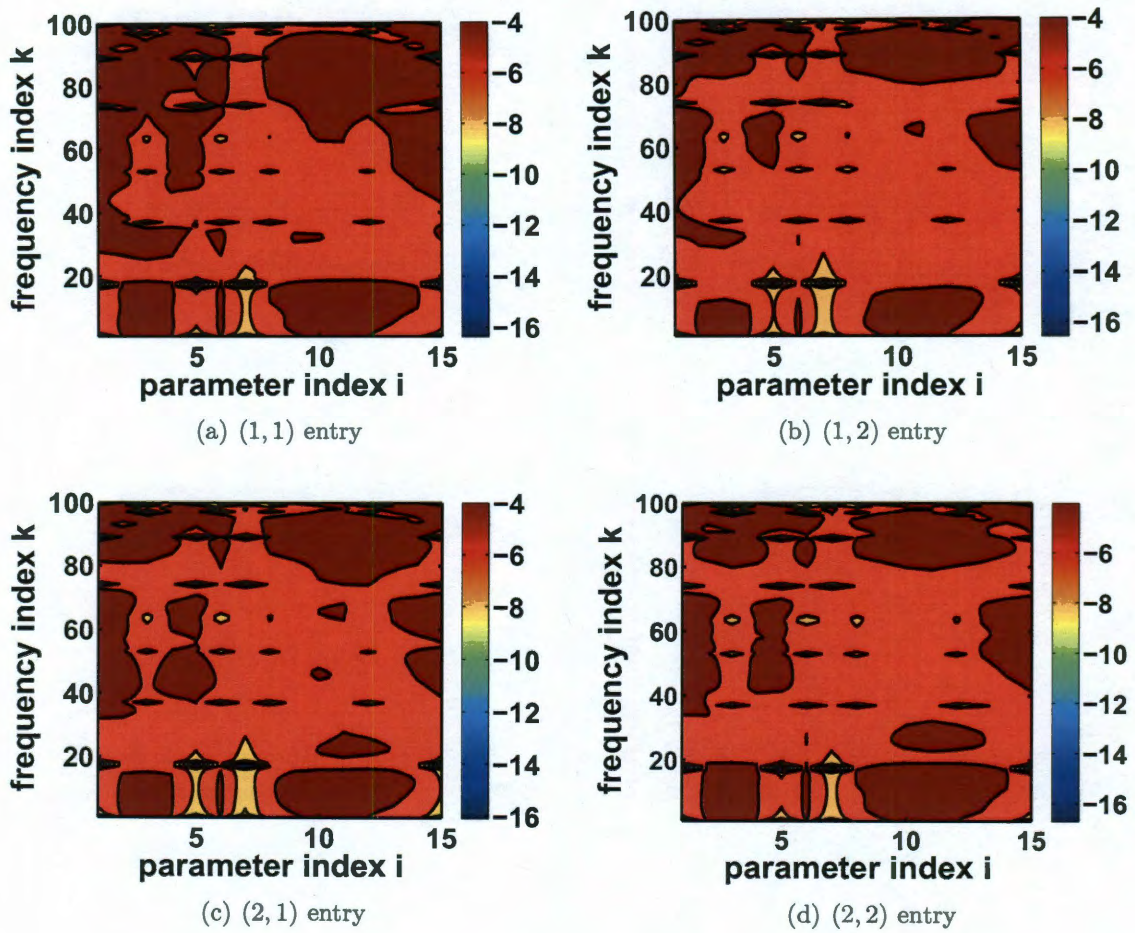


Figure 5.9: Errors for all frequencies and parameters

This reduced model is able to predict the behavior of the 2-port S-parameters within an error of 50dB for all 200 measurements of  $s$  and 15 values of the parameter  $w$ .

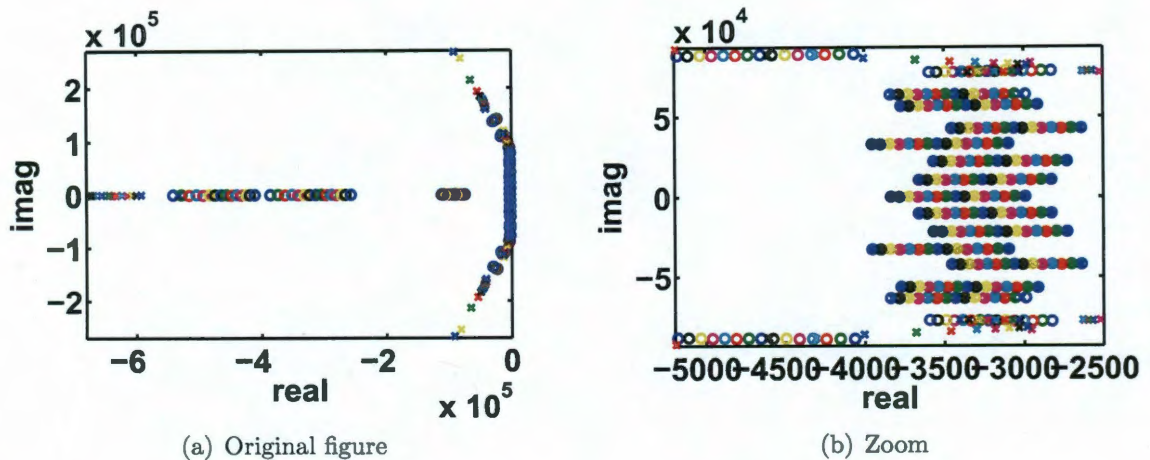


Figure 5.10: Poles after modeling vs. recovered poles - 'o': original, '+' : recovered

We also compared the poles with respect to the frequency, obtained by substituting each parameter value in our realization, to those obtained after the modeling step (shown initially in Fig. 5.7). After removing poles with large magnitude which are, in fact, poles at infinity, but due to numerical issues appear as 5 orders of magnitude higher than the rest, we obtain the plot in Fig. 5.10. We notice that all recovered poles (obtained from the final realization) are stable, even though they are not precisely in the same location as the original ones (obtained after the modeling step). This is due to the fact that we are dealing with measurement noise and the two variable MIMO Loewner matrix is not precisely singular.

Fig. 5.11 shows the magnitude of all entries for parameter value  $w_1 = 60\mu m$ . We plot the measured data, as well as the model on a dB scale.

Instead of looking at each entry, an alternative way to checking the accuracy of the model is to look at the sigma plot. Fig. 5.12 shows the 2 singular values of each measured S-parameter, as well as the 2 singular values of the matrix we obtain by evaluating our model at  $j2\pi f$  for each frequency (this is the same as one would obtain



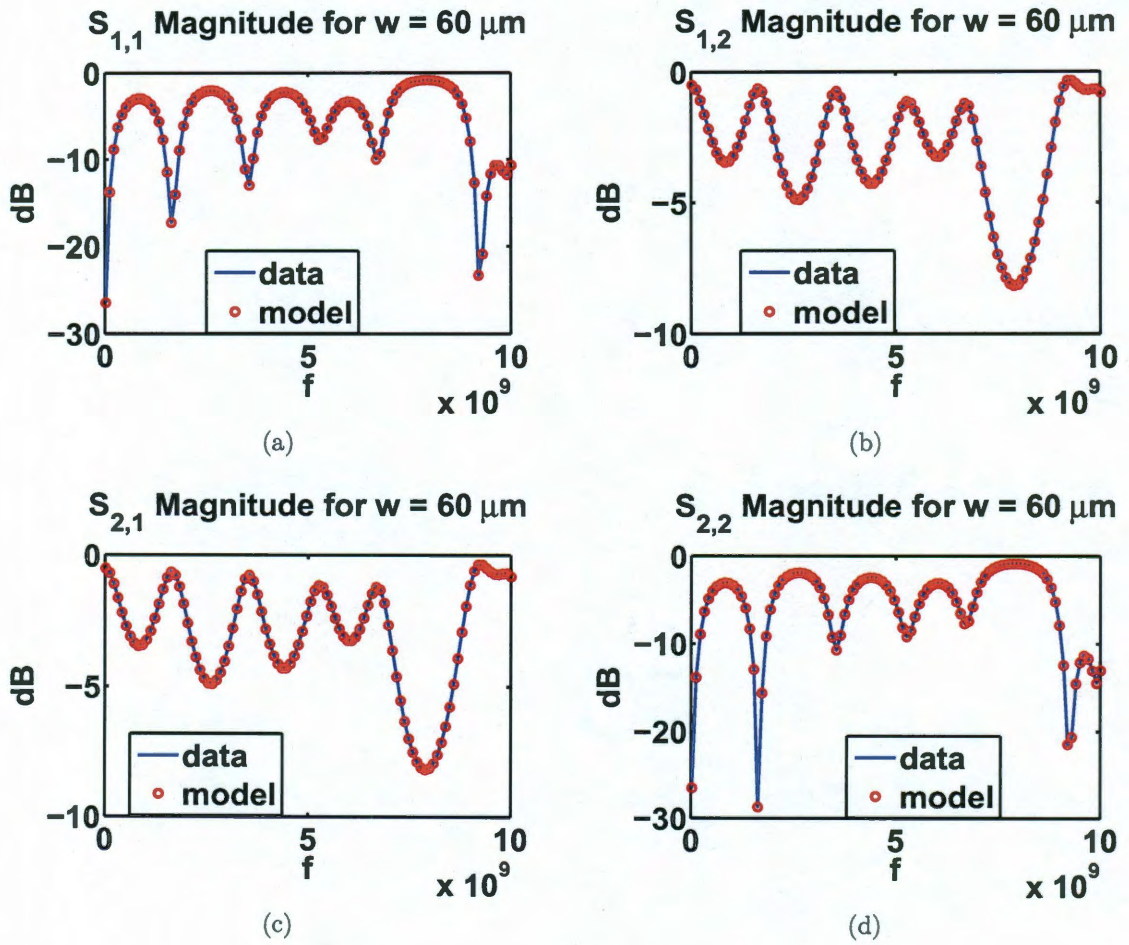


Figure 5.11: Magnitude plots

by using the function `sigma` in Matlab and focus on the desired frequency range). We are only presenting the plots for parameter values  $w_1 = 60\mu m$  and  $w_{15} = 130\mu m$  since they yield the largest errors.

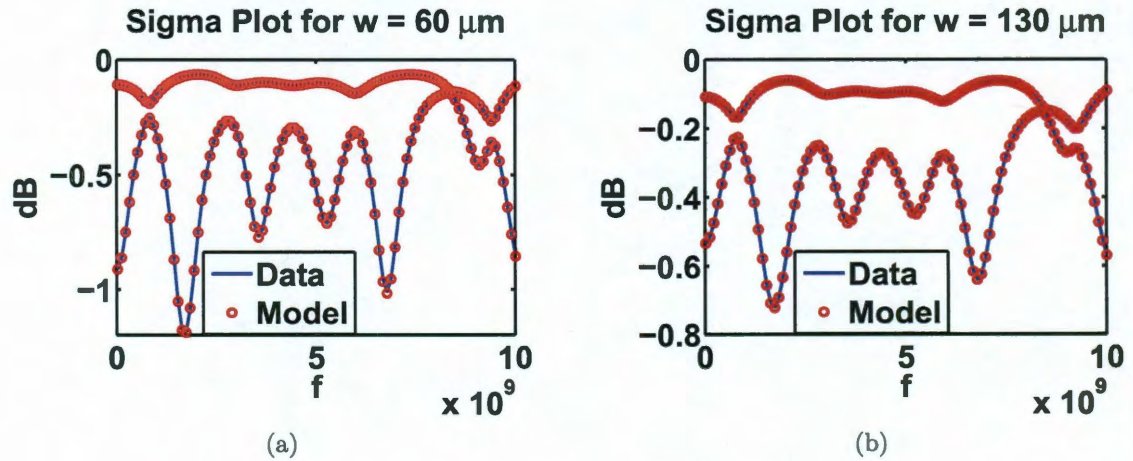


Figure 5.12: Frequency response plots

Fig. 5.13 shows the errors in each entry for parameter value  $w_1 = 60\mu m$  on a dB scale. Indeed, errors are below  $-50dB$ , as expected. Alternatively, a compressed way to obtain information about the errors is through the 2 singular values of the matrix obtained by subtracting the measured S-parameter from the S-parameter estimated with our model. Fig. 5.14 shows the singular values of the error matrices, for parameter values  $w_1 = 60\mu m$  and  $w_{15} = 130\mu m$ , which yield the largest errors.

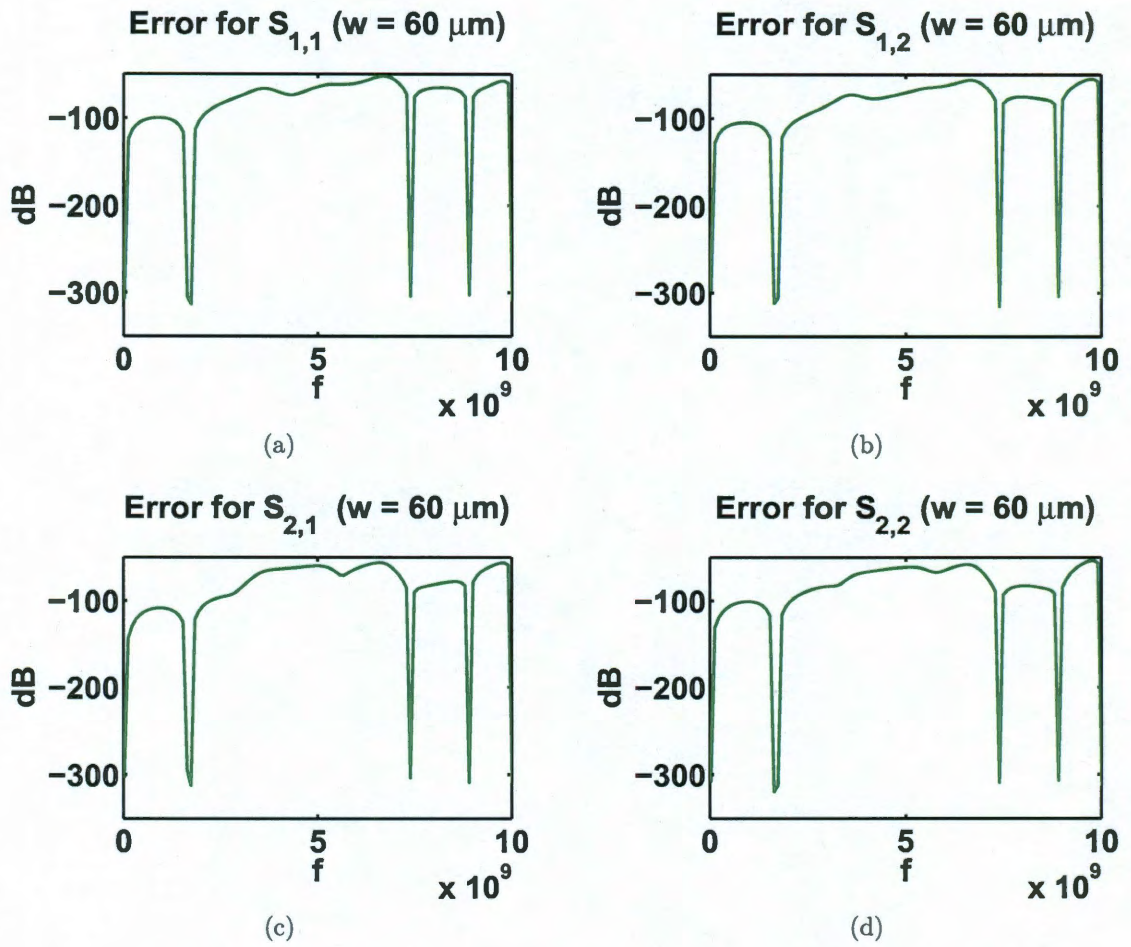


Figure 5.13: Error plots

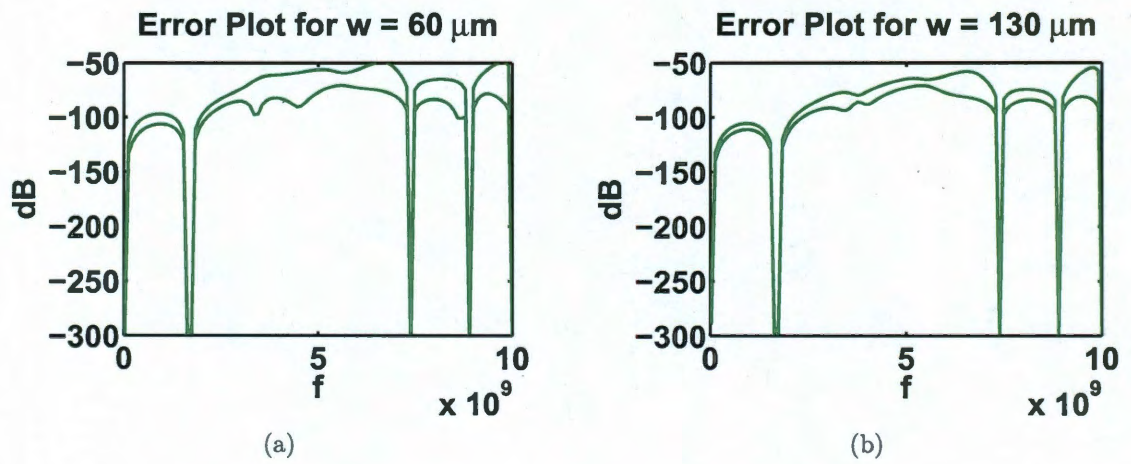


Figure 5.14: Compressed error plots

### 5.5.3 Convection-Diffusion Equation

Consider the 2-dimensional convection-diffusion equation in the domain  $\Omega = (0, 1)^2$

$$\frac{\partial \mathbf{x}}{\partial t}(t, \xi) = \Delta \mathbf{x}(t, \xi) + p \cdot \nabla \mathbf{x}(t, \xi) + \mathbf{b}(\xi) \mathbf{u}(t), \xi \in \Omega,$$

$$\mathbf{x}(t, \xi) = \mathbf{0}, \xi \in \partial\Omega,$$

which models one-dimensional convective transport. The parameter  $p \in [0, 10]$  is time-invariant. A finite-difference discretization with 30 points in each spatial coordinate leads to an order  $n = 900$  parametric system:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + p\mathbf{A}_1\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \mathbf{x}(0) = \mathbf{0},$$

where the input matrix is  $\mathbf{B} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T$  and the output matrix is  $\mathbf{C} = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}$ . The resulting transfer function is

$$\mathbf{H}(s, p) = \mathbf{C}(s\mathbf{I} - \mathbf{A} - p\mathbf{A}_1)^{-1}\mathbf{B},$$

which shows that the complexity in the parameter,  $m$  is also 900. We first select 50 measurements in the Laplace variable  $s$  logarithmically distributed on the imaginary axis between  $j10^{-2}$  and  $j10^6$  and 30 measurements in the parameter  $p$  linearly distributed between 0 and 10. Using only 8 of these measurements with respect to the frequency, as well as 8 measurements with respect to the parameter, we are able to construct a reduced system of order  $n = m = 7$ . Using these measurements, the singular value drop of the 2D Loewner matrix is shown in Fig. 5.15 (we notice that the matrix is indeed singular, so using the vector in its right nullspace, we are able to construct a realization).

When evaluating the reduced model on the initial grid of 50 frequencies and 30

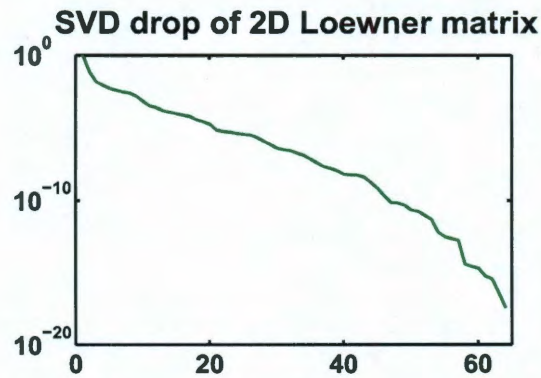
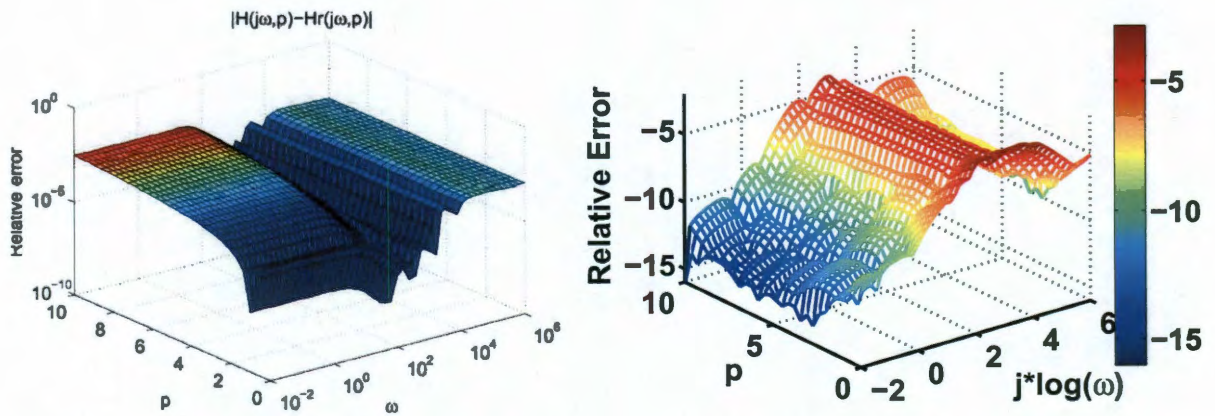


Figure 5.15: Singular value drop of the 2D Loewner matrix

parameter values and computing the difference with respect to the original measurements, we obtain a maximum relative error of  $1.3e - 3$  (Fig. 5.16(b)). Fig. 5.16(a) shows the relative error for a reduced system of order  $n = m = 8$  obtained with the method proposed in [77], for which the maximum error on the grid is  $3.1e - 3$ .



(a) Errors obtained with the method described in [77]

(b) Errors obtained with our approach

Figure 5.16: Relative errors

## Convergence of vector fitting

---

This chapter investigates the convergence behavior of vector fitting. The analysis starts with a summary of the main steps of the vector fitting (VF) [18] algorithm. It is recognized that the pole relocation iteration is, in fact, finding the roots of a set of coupled multivariable rational equations. For noise-free measurements, it is shown that there is no iteration involved, assuming that the number of starting poles is chosen greater or equal to the order of the underlying system. For high noise ratios, the vector fitting iteration may not find any solution due to the fact that all fixed points of the iteration are repelling. Therefore, we propose to incorporate the Newton step in the VF iteration in case the iteration does not converge, to guarantee its convergence. The main contribution of this chapter is the convergence analysis of the pole relocation process of the vector fitting procedure.

### 6.1 Review of vector fitting

In the sequel we provide a short review of the basic ideas behind vector fitting (VF) [18], [19], [22], [21], [23]. We assume that the measurements are generated by a linear time-invariant dynamical system with transfer function  $H(s)$ . When modeling a single input single output (SISO) system, VF aims at finding an approximation

$f(s)$  of its transfer function expressed in pole residue form

$$H(s) \approx f(s), \quad f(s) = \sum_{i=1}^k \frac{r_i}{s - p_i} + d. \quad (6.1)$$

The finite residues  $r_i$  and poles  $p_i$  are either real quantities or come in complex conjugate pairs, while  $d$  is real. In the case of noise-free measurements, the goal is to identify the underlying system, meaning that the poles and residues of the model  $f(s)$  should match the ones of the original system  $H(s)$ . The problem expressed in Eq. (6.1) is nonlinear in the poles  $p_i$ . Vector fitting solves it iteratively as a sequence of linear problems in two stages.

**Stage 0: order selection**

Decide a priori on an approximation order  $n$ .

**Stage 1: pole identification**

Assume the usual pole-residue representation for the numerator  $\mathbf{n}(s)$  and denominator  $\mathbf{d}(s)$  of the unknown function  $f(s)$ :

$$\begin{bmatrix} \mathbf{n}(s) \\ \mathbf{d}(s) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n \frac{c_i}{s - a_i} + d \\ \sum_{i=1}^n \frac{\tilde{c}_i}{s - a_i} + 1 \end{bmatrix}. \quad (6.2)$$

Note that  $\mathbf{n}(s)$  and  $\mathbf{d}(s)$  share the same poles. Multiplying the second row in Eq. (6.2) by  $f(s)$  yields the usual linearization:

$$f(s) = \frac{\mathbf{n}(s)}{\mathbf{d}(s)} \Rightarrow \underbrace{\sum_{i=1}^n \frac{c_i}{s - a_i} + d}_{\mathbf{n}(s)} = \underbrace{\left( \sum_{i=1}^n \frac{\tilde{c}_i}{s - a_i} + 1 \right)}_{\mathbf{d}(s)} f(s). \quad (6.3)$$

Eq. (6.3) is linear in its unknowns  $c_i$ ,  $d$ ,  $\tilde{c}_i$ , but nonlinear in the quantities  $a_i$ . By specifying a set of poles  $a_i$ , this equation becomes linear. Next, the fact that  $f(s)$  interpolates the original transfer function  $H(s)$  at the measurement samples is used

to write Eq. (6.3) for the sample points at which the measurements are provided.

This leads to an overdetermined linear system

$$\mathbf{A}(s_i, N; a_i, n) \mathbf{x}(c_i, n; d; \tilde{c}_i, n) = \mathbf{b}, \text{ where} \quad (6.4)$$

$$\mathbf{A}(s_i, N; a_i, n) = \left[ \Phi(s_i, N; a_i, n) \quad \mathbf{1} \quad -\text{diag}(\mathbf{b})\Phi(s_i, N; a_i, n) \right] \in \mathbb{C}^{N \times (2n+1)}, \text{ with} \quad (6.5)$$

$$\Phi(s_i, N; a_i, n) = \begin{bmatrix} \frac{1}{s_1 - a_1} & \cdots & \frac{1}{s_1 - a_n} \\ \vdots & \ddots & \vdots \\ \frac{1}{s_N - a_1} & \cdots & \frac{1}{s_N - a_n} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} H_1 \\ \vdots \\ H_N \end{bmatrix} \in \mathbb{C}^{N \times 1} \quad (6.6)$$

$$\mathbf{x}(c_i, n; d; \tilde{c}_i, n) = \begin{bmatrix} \mathbf{c} \\ d \\ \tilde{\mathbf{c}} \end{bmatrix} \in \mathbb{C}^{(2n+1) \times 1}, \text{ where } \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \quad \tilde{\mathbf{c}} = \begin{bmatrix} \tilde{c}_1 \\ \vdots \\ \tilde{c}_n \end{bmatrix}, \quad (6.7)$$

with  $s_i \in \mathbb{C}$  and  $H_i = H(s_i) \in \mathbb{C}$ , for  $i = 1, \dots, N$ . The matrix  $\Phi(s_i, N; a_i, n)$  is a Cauchy matrix. By fixing  $a_i$ , the unknowns are solely in the solution vector  $\mathbf{x}$ , which is found by solving the overdetermined linear system in Eq. (6.4) as a least squares problem.

Rewriting Eq. (6.2) as a pole-zero expansion, we obtain:

$$\mathbf{n}(s) = d \frac{\prod_{i=1}^n (s - z_i)}{\prod_{i=1}^n (s - a_i)} \quad (6.8)$$

$$\mathbf{d}(s) = \frac{\prod_{i=1}^n (s - \bar{z}_i)}{\prod_{i=1}^n (s - a_i)}, \quad (6.9)$$



so  $f(s)$ , the approximation of the underlying  $H(s)$ , can be computed as

$$f(s) = \frac{\mathbf{n}(s)}{\mathbf{d}(s)} = d \frac{\prod_{i=1}^n (s - z_i)}{\prod_{i=1}^n (s - \bar{z}_i)}. \quad (6.10)$$

This shows that the zeros of the expression in the denominator  $\mathbf{d}(s)$  become the poles of  $f(s)$ . The zeros  $\bar{z}_i$  are found by solving an eigenvalue problem:

$$\mathbf{M}\mathbf{v}_i = \bar{z}_i\mathbf{v}_i, \text{ where } \mathbf{M} = \text{diag}(\mathbf{a}) - \mathbf{1}\bar{\mathbf{c}}^T, \text{ with } \mathbf{a}^T = \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix}. \quad (6.11)$$

The vector  $\mathbf{a}$  contains the current set of poles, while  $\bar{\mathbf{c}}$  is found from the least squares solution in Eq. (6.7). Now, using the zeros  $\bar{z}_i$  as the new poles  $a_i$ , a new linear system like in Eq. (6.4) is set up and the vector fitting iteration repeats the procedure described up to this point for a certain number of steps, a number on which the user decides. This is referred to as the pole relocation process.

### Stage 2: residue identification

One can solve for the residues  $r_i$  and the asymptotic term  $d$  using the original problem in Eq. (6.1) using the already determined poles of  $f(s)$  ( $p_i = a_i$ ). This leads to another overdetermined linear system  $\mathbf{P}(s_i, N; p_i, n)\mathbf{y}(r_i, n; d) = \mathbf{b}$ :

$$\mathbf{P}(s_i, N; p_i, n) = \begin{bmatrix} \Phi(s_i, N; p_i, n) & \mathbf{1} \end{bmatrix} \in \mathbb{C}^{N \times (n+1)}, \quad (6.12)$$

$$\mathbf{y}(r_i, n; d) = \begin{bmatrix} \mathbf{r} \\ d \end{bmatrix} \in \mathbb{C}^{(n+1) \times 1}, \quad \mathbf{r} = \begin{bmatrix} r_1 & \dots & r_n \end{bmatrix}^T, \quad (6.13)$$

which is also solved as a least squares problem.

### 6.1.1 Vector fitting for MIMO systems

First, we review the case of single-input multiple-output systems. When fitting a vector-valued function for which measurements  $\mathbf{H}_i \in \mathbb{C}^{p \times 1}$ , for  $i = 1, \dots, N$ , are provided, the matrices in Eq.(6.5) - (6.7) change to:

$$\mathbf{A}(s_i, N; a_i, n) = \begin{bmatrix} \Phi_v & \mathbf{1}_v & -diag(\mathbf{b}_v)\Phi_v \end{bmatrix}, \text{ where} \quad (6.14)$$

$$\Phi_v = \begin{bmatrix} \frac{\mathbf{I}_p}{s_1 - a_1} & \cdots & \frac{\mathbf{I}_p}{s_1 - a_n} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{I}_p}{s_N - a_1} & \cdots & \frac{\mathbf{I}_p}{s_N - a_n} \end{bmatrix}, \mathbf{1}_v = \begin{bmatrix} \mathbf{I}_p \\ \vdots \\ \mathbf{I}_p \end{bmatrix}, \mathbf{b}_v = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_N \end{bmatrix}, \quad (6.15)$$

with  $s_i \in \mathbb{C}$ ,  $\mathbf{H}_i = \mathbf{H}(s_i) \in \mathbb{C}$ , for  $i = 1, \dots, N$ , and  $\mathbf{I}_p$ , the identity matrix of dimension  $p$ . For vector valued functions, the residues of the numerator, namely  $c_i$ , are of dimension  $p \times 1$ , while the ones in the denominator, namely  $\tilde{c}_i$  are scalars. The dimensions of the matrices in Eq.(6.14)-(6.15) are  $\mathbf{A} \in \mathbb{C}^{(Np) \times (n(p+1)+p)}$  and  $\mathbf{b}_v \in \mathbb{C}^{(Np) \times 1}$ , while the solution vector  $\mathbf{x} \in \mathbb{C}^{(n(p+1)+p) \times 1}$ . After performing row permutations (which correspond to changing the order of the equations), we obtain (assuming  $\mathbf{d} = \mathbf{0}$ ) [25]

$$\begin{bmatrix} \Phi & & -diag(H_1)\Phi \\ & \ddots & \vdots \\ & & \Phi & -diag(H_p)\Phi \end{bmatrix} \begin{bmatrix} C_1 \\ \vdots \\ C_p \\ \tilde{c} \end{bmatrix} = \begin{bmatrix} H_1 \\ \vdots \\ H_p \end{bmatrix}, \quad (6.16)$$

where

$$H_i = \begin{bmatrix} H_i(s_1) \\ \vdots \\ H_i(s_N) \end{bmatrix}, C_i = \begin{bmatrix} c_{1,i} \\ \vdots \\ c_{N,i} \end{bmatrix}, i = 1, \dots, p,$$

and  $H_i(s)$  is the  $i^{\text{th}}$  entry of the vector valued transfer function  $\mathbf{H}(s) = \begin{bmatrix} H_1(s) & \dots & H_p(s) \end{bmatrix}^T$ . This expression is the one implemented in the vector fitting software.

The linear system presented in Eq. (6.12)-(6.13) changes accordingly and can be rewritten similarly to Eq. (6.16) after appropriate row permutations.

The theory presented up to now assumes a single-input single-output or a single-input multiple-output system. In the case of multiple-input multiple-output systems, fitting can be performed matrix-wise, column-wise or element-wise.

1. Since vector fitting works for scalar or vector-valued rational functions, *matrix-wise fitting* stacks all the columns into a column vector  $\text{vec}(\mathbf{H}(s))$  and fits this with the measurements  $\text{vec}(\mathbf{H})$  [46]. As detailed in [23], for matrix-wise fitting, all entries of the transfer function share the same poles and the resulting state-space realization is of dimension  $n \cdot p$ , where  $n$  is the number of starting poles, when no  $\mathbf{D}$  term is required.
2. For *column-wise fitting*, each column of the transfer function is fitted by a common set of poles and the resulting state-space realization is, again, of dimension  $n \cdot p$  (the same starting poles are used for all columns).
3. For *element-wise fitting*, different sets of poles are obtained for each entry of the transfer function, and the resulting state-space realization is of dimension  $n \cdot p^2$  (the same starting poles are used for all entries).

For all the numerical examples presented in Sect. 4.4, vector fitting was used to fit MIMO systems with column-wise fitting. Even though [23] has shown experimentally that element-wise fitting is faster and yields smaller errors, the order of the resulting models will be too large, as they are multiples of the second power of the number of ports  $p$ . Therefore, for a data set obtained from a device with  $p = 50$  ports, the smallest dimension of the resulting model built with the element-wise implementation of VF is  $p^2 = 2500$  (when using only one pole to fit each entry). The main goal in

model reduction is constructing models of the least possible order which approximate the frequency response in a certain frequency band. Since models of large dimensions are expensive to use in future simulations, we decided to use column-wise fitting instead of element-wise fitting [46].

### 6.1.2 Issues with vector fitting

1. As pointed out in [21], there is no proof on when or how fast vector fitting converges. The goal of Sect. 6.3 is to elucidate this problem and provide insight into the convergence properties of the vector fitting iteration.
2. The convergence rate of the pole relocation process strongly depends on the starting poles. Experimentally, it has been shown in [18] that choosing the starting poles as either real numbers linearly distributed in the desired frequency range, or as complex conjugate pairs with the imaginary part linearly distributed in the desired frequency range and the real part as 1% of the imaginary part is a heuristic which yields good results for resonant systems. Sect. 6.7 shows that one might choose them as some of the measurements due to the fact that interpolation is exact at the points  $a_i$ .
3. Unstable poles may result during the iterative process due to the fact that, at iteration step  $k$ , the poles of the approximating rational function are the zeros computed at iteration step  $k-1$  and the zeros of a function may be anywhere in the complex plane. To avoid having models which are unstable, it was proposed in [18] to either remove the unstable poles or flip the sign of their real part to move them in the left-half plane.
4. The residue identification stage is not needed when no pole flipping is involved, due to Lemma 6.1.1.

**Lemma 6.1.1.** *The following descriptor realization of  $f(s)$  can be obtained from the quantities found in the pole identification stage:  $f(s) = \mathbf{C}\Phi(s)^{-1}\mathbf{B}$ , where*

$$\mathbf{C} = \begin{bmatrix} \mathbf{d} & \mathbf{c}_1 & \dots & \mathbf{c}_n \end{bmatrix}, \quad \Phi(s) = \begin{bmatrix} 1 & a_1 - s & \dots & & \\ \vdots & & \ddots & & \\ 1 & & & a_n - s & \\ 1 & \tilde{c}_1 & \tilde{c}_2 & \dots & \tilde{c}_n \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{n \times 1} \\ 1 \end{bmatrix}, \quad (6.17)$$

where  $\mathbf{d}, \mathbf{c}_1, \dots, \mathbf{c}_n \in \mathbb{C}^{p \times 1}$  are found by solving Eq. (6.14)-(6.15). This formula is given for the general case of vector transfer functions (1 input,  $p$  outputs).

*Proof.* Let us denote  $\Phi^{-1}(s)\mathbf{B}$  as  $\mathbf{x}$ , so  $\Phi(s)\mathbf{x} = \mathbf{B}$ :

$$\begin{bmatrix} 1 & a_1 - s & \dots & & \\ \vdots & & \ddots & & \\ 1 & & & a_n - s & \\ 1 & \tilde{c}_1 & \tilde{c}_2 & \dots & \tilde{c}_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (6.18)$$

$$\Rightarrow \begin{cases} \mathbf{x}_1 = \frac{1}{s-a_1}\mathbf{x}_0, & \text{if } s \neq a_1, \\ \vdots \\ \mathbf{x}_n = \frac{1}{s-a_n}\mathbf{x}_0, & \text{if } s \neq a_n, \\ \mathbf{x}_0 + \tilde{c}_1\mathbf{x}_1 + \dots + \tilde{c}_n\mathbf{x}_n = 1 \Rightarrow \left(1 + \tilde{c}_1\frac{1}{s-a_1} + \dots + \tilde{c}_n\frac{1}{s-a_n}\right)\mathbf{x}_0 = 1. \end{cases}$$

Since

$$\begin{aligned} f(s) &= \mathbf{C} \underbrace{\Phi(s)^{-1}\mathbf{B}}_{\mathbf{x}} = \mathbf{C}\mathbf{x} = \mathbf{d}\mathbf{x}_0 + \mathbf{c}_1\mathbf{x}_1 + \dots + \mathbf{c}_n\mathbf{x}_n = \left(\mathbf{d} + \frac{\mathbf{c}_1}{s-a_1} + \dots + \frac{\mathbf{c}_n}{s-a_n}\right)\mathbf{x}_0 \\ &= \frac{\mathbf{d} + \frac{\mathbf{c}_1}{s-a_1} + \dots + \frac{\mathbf{c}_n}{s-a_n}}{1 + \frac{\tilde{c}_1}{s-a_1} + \dots + \frac{\tilde{c}_n}{s-a_n}} = \frac{\mathbf{n}(s)}{\mathbf{d}(s)}, \end{aligned} \quad (6.19)$$

which is precisely as expected from Eq.(6.3).  $\square$

5. The denominator  $\mathbf{d}(s)$  is forced to approach unity at high frequencies in [18], while in [19] this condition has been eliminated, leading to the variant called Relaxed Vector Fitting.

### 6.1.3 Vector fitting as a Sanathanan-Koerner iteration

Vector fitting was recognized as a Sanathanan-Koerner (SK) iteration [28]. The goal of SK is to solve the nonlinear minimization problem [78]

$$\min \left\| \frac{\mathbf{N}(s)}{D(s)} - \mathbf{H}(s) \right\|, \quad (6.20)$$

with  $\mathbf{N}(s)$ , the numerator, and  $D(s)$ , the denominator polynomial of the transfer function. After linearization and multiplication by a weighting factor  $\frac{1}{D^{(k-1)}(s)}$ , we obtain the following expression

$$\min \left\| \frac{\mathbf{N}^{(k)}(s)}{D^{(k-1)}(s)} - \frac{D^{(k)}(s)}{D^{(k-1)}(s)} \mathbf{H}(s) \right\|, \quad (6.21)$$

which is solved via an iteration. The superscript  $^{(k)}$  denotes the quantity at iteration  $k$ . By introducing the notation  $\frac{\mathbf{N}^{(k)}(s)}{D^{(k)}(s)} \frac{D^{(k)}(s)}{D^{(k-1)}(s)} = \mathbf{n}^{(k)}(s)$  and  $\frac{D^{(k)}(s)}{D^{(k-1)}(s)} = \mathbf{d}^{(k)}(s)$ , we obtain the relationship

$$\mathbf{n}^{(k)}(s) - \mathbf{d}^{(k)}(s) \mathbf{H}(s) \approx \mathbf{0}, \quad (6.22)$$

which is precisely Eq. (6.3). The weighting factor  $\frac{1}{D^{(k-1)}(s)}$  is in fact

$$\frac{1}{D^{(k-1)}(s)} = \frac{\prod_{i=1}^n (s - \bar{z}_i)}{\prod_{i=1}^n (s - a_i)},$$

where  $a_i$  are the current starting poles and  $\bar{z}_i$  are the poles at the new iteration computed from the eigenvalue problem in Eq. (6.11). At convergence, these quantities will be equal, so the weighting will be 1.

## 6.2 Solution to vector fitting using VARPRO

As we shall see in Sect. 6.3, the vector fitting iteration may fail to converge for noisy data, therefore this section proposes a solution to VF using VARPRO. Variable projection (VARPRO) [29, 30] is a method for solving the separable non-linear least squares problems. In particular, it can be used to solve our problem when regarded as a minimization problem:

$$\min \sum_i \|H_i - f(s_i)\|^2, \quad f(s) = \sum_{j=1}^n \frac{r_j}{s - p_j} + d. \quad (6.23)$$

Since there is no linearization involved (as it is the case in VF, as one can see from Eq. (6.3)), VARPRO is expected to yield better results in the case of noisy measurements since the original minimization problem is solved without introducing any bias.

The goal is to minimize the sum of the norm of the deviation between the  $i^{\text{th}}$  measurement and the model evaluated at the  $i^{\text{th}}$  sample point:

$$e_i(\mathbf{r}, \mathbf{p}) = H_i - \sum_{j=1}^n r_j \phi_j(s_i), \quad \text{with } \phi_j(s_i) = \frac{1}{s_i - p_j}. \quad (6.24)$$

The problem consists in determining the linear coefficients  $r_j$  (the residues) and the nonlinear parameters  $p_j$  (the poles) to minimize the norm of the error vector:

$$E = \|e(\mathbf{r}, \mathbf{p})\|_2^2 = \|\mathbf{b} - \Phi(s_i, N; p_i, n)\mathbf{r}\|_2^2, \quad (6.25)$$

where  $\Phi(s_i, N; p_i, n)$  is the Cauchy matrix defined in Eq. (6.6). Clearly, if the nonlinear parameters  $p_i$  were known, the linear parameters  $r_j$  would be determined by solving Eq. (6.25) as a linear least squares problem:

$$\mathbf{r} = \Phi(s_i, N; p_i, n)^+ \mathbf{b},$$

where  $\Phi(s_i, N; p_i, n)^+$  is the Moore-Penrose generalized inverse of  $\Phi$  (for simplicity,  $\Phi$ , instead of  $\Phi(s_i, N; p_i, n)$  will be used in the sequel). Replacing this expression of  $\mathbf{r}$  into Eq. (6.25) leads to the following minimization problem, which is only in terms of the nonlinear parameters, as the linear ones have been eliminated:

$$\min_{\mathbf{p}} \frac{1}{2} \|(\mathbf{I} - \Phi\Phi^+)\mathbf{b}\|_2^2.$$

The quantity  $e_2(\mathbf{p}) = (\mathbf{I} - \Phi\Phi^+)\mathbf{b}$  is referred to as the variable projection residual. The reason for the terminology is the fact that the matrix  $\mathbf{I} - \Phi\Phi^+$  represents the projector on the orthogonal complement of the column space of  $\Phi$  [30].

**Theorem 6.2.1.** [29] *Assuming that in the open set  $\Omega \subset \mathbb{R}$ , the matrix  $\Phi$  has constant rank  $r \leq \min(N, n)$ , then*

1. *if  $\hat{\mathbf{p}}$  is a critical point (or global minimizer) for  $e_2(\mathbf{p})$  and  $\mathbf{r}$  satisfies  $\mathbf{r} = \Phi^+\mathbf{b}$ , then  $\hat{\mathbf{p}}$  is a critical point of  $e(\mathbf{r}, \mathbf{p})$  and  $e(\mathbf{r}, \mathbf{p}) = e_2(\mathbf{p})$ ;*
2. *if  $(\hat{\mathbf{r}}, \hat{\mathbf{p}})$  is a global minimizer of  $e(\mathbf{r}, \mathbf{p})$  then  $\hat{\mathbf{p}}$  is a global minimizer of  $e_2(\mathbf{p})$  and  $e_2(\mathbf{p}) = e(\mathbf{r}, \mathbf{p})$ . Furthermore, if there is an unique  $\hat{\mathbf{r}}$  among the minimizing pairs of  $e(\mathbf{r}, \mathbf{p})$ , then  $\hat{\mathbf{r}}$  must satisfy  $\mathbf{r} = \Phi^+\mathbf{b}$ .*

Solving  $\min_{\mathbf{p}} \frac{1}{2} \|e_2(\mathbf{p})\|_2^2$  is achieved by considering its gradient, which is expressed in terms of the Jacobian  $\mathbf{J}_{e_2(\mathbf{p})}$ . The Jacobian is the matrix obtained by taking partial derivatives of the vector  $e_2(\mathbf{p})$  with respect to each unknown in the vector  $\mathbf{p}$ . The gradient is

$$\nabla \frac{1}{2} \|e_2(\mathbf{p})\|_2^2 = (\mathbf{J}_{e_2(\mathbf{p})})^T e_2(\mathbf{p})$$

and the  $j^{\text{th}}$  column of the Jacobian can be written as

$$\mathbf{J}_{e_2(\mathbf{p})}(:, j) = - \left[ \left( \mathbf{P}_{\Phi}^{\perp} \frac{\partial \Phi}{\partial p_j} \Phi^{-} \right) + \left( \mathbf{P}_{\Phi}^{\perp} \frac{\partial \Phi}{\partial p_j} \Phi^{-} \right)^T \right] \mathbf{b},$$



where  $\Phi^-$  is a symmetric generalized inverse satisfying  $\Phi\Phi^-\Phi = \Phi$  and  $(\Phi\Phi^-)^T = \Phi\Phi^-$  (the full Moore-Penrose pseudoinverse is not necessary) and  $\mathbf{P}_\Phi^\perp = \mathbf{I} - \Phi\Phi^-$ .

The Gauss-Newton iteration with step control which solves the non-linear least squares problem

$$\min_{\mathbf{p}} e_2(\mathbf{p}) = \min_{\mathbf{p}} \|e_2(\mathbf{p})\|^2 = \min_{\mathbf{p}} \|\mathbf{P}_\Phi^\perp \mathbf{b}\|^2$$

is

$$\mathbf{p}_{k+1} \leftarrow \mathbf{p}_k - t_k (\mathbf{J}_{e_2}(\mathbf{p}_k))^{-1} e_2(\mathbf{p}_k),$$

where  $t_k > 0$  is a parameter controlling the size of the step. If  $e_2(\mathbf{p}_{k+1}) < e_2(\mathbf{p}_k)$ , then  $t_k = 1$ , otherwise  $t_k$  is chosen such that  $e_2(\mathbf{p}_{k+1}) < e_2(\mathbf{p}_k)$  (steepest descent). To ensure convergence, it is necessary to adjust the size of the step.

For MIMO systems, Eq. (6.25) needs to be solved for each input-output pair, therefore leading to a least squares problem with multiple right-hand sides [79].

### 6.3 Analysis of the pole relocation iteration in vector fitting

In this section we consider the original vector fitting formulation with the denominator of the form  $\mathbf{d}(s) = \sum_{i=1}^n \frac{\tilde{c}_i}{s-a_i} + 1$ . We denote the starting poles  $a_i$  at iteration  $k$  as  $a_i^{(k)}$ . Convergence is achieved when the poles  $a_i$  stay unchanged ( $a_i^{(k)} = a_i^{(k+1)}$ ).

**Theorem 6.3.1.** *Vector fitting seeks to find the roots of a system of coupled rational equations  $\tilde{\mathbf{c}}(\mathbf{a}) = \mathbf{0}$ , where  $\tilde{\mathbf{c}}$  and  $\mathbf{a}$  are vectors of dimension  $n$ , the approximation order of the model. Therefore, the VF iteration converges when  $\tilde{c}_i = 0, i = 1, \dots, n$ .*

**Remark 6.3.1.** *At convergence, the denominator  $\mathbf{d}(s)$  becomes 1 and the numerator  $\mathbf{n}(s)$  provides the pole-residue expansion we are seeking, so the second step, namely the residue identification stage, is not needed.*

**Remark 6.3.2.** *Solutions of a system of coupled multivariable rational equations can be computed approximately through iterative methods, like the Newton fixed point iteration. The latter has the advantage of converging if the initial guess is close enough to the solution.*

**Corollary 6.3.1.** *Provided that the measurements are noise-free, the number of measurements is sufficient, and the number of starting poles is greater than the order of the underlying system, convergence occurs at step 2.*

*Proof of Corollary 6.3.1.* When using at least as many starting poles as the order of the underlying system, the number of measurements is sufficient and the measurements are noise-free, the following relationship holds with equality (for noisy measurements is it only an approximation):

$$f(s) = \frac{\mathbf{n}(s)}{\mathbf{d}(s)} \Rightarrow \underbrace{\sum_{i=1}^n \frac{c_i}{s - a_i}}_{\mathbf{n}(s)} + d = \underbrace{\left( \sum_{i=1}^n \frac{\tilde{c}_i}{s - a_i} + 1 \right)}_{\mathbf{d}(s)} f(s). \quad (6.26)$$

At the first step, the coefficients  $c_i$ ,  $d$ ,  $\tilde{c}_i$  are found from the linear system in Eq.(6.4) and they are simply the residues of the numerator and denominator expressed in the rational basis  $\left\{ \frac{1}{s-a_1}, \dots, \frac{1}{s-a_n} \right\}$ . At step 2, the quantities  $a_i$  are considered as the zeros of the denominator  $\mathbf{d}(s)$  at the current step. Therefore,  $a_i^{(2)}$  are the poles of the underlying system (recall that the poles of a system are defined as the zeros of the polynomial in the denominator). ■

*Proof of Theorem 6.3.1.* At convergence, we have that  $a_i^{(k)} = a_i^{(k+1)}$ , where  $a_i^{(k)}$  are the poles at the current iteration and  $a_i^{(k+1)}$  are the zeros of the denominator  $\mathbf{d}(s)$ :

$$\mathbf{d}(s) = \frac{\prod_{i=1}^n (s - a_i^{(k+1)})}{\prod_{i=1}^n (s - a_i^{(k)})}$$

Therefore, at convergence, we have pole-zero cancellations, so  $\mathbf{d}(s) = 1$  and, consequently,  $\tilde{c}_i = 0, i = 1, \dots, n$ .

Multiplying Eq.(6.26) by  $\prod_{j=1}^n (s - a_j)$  yields

$$\sum_{i=1}^n c_i \prod_{j=1, j \neq i}^n (s - a_j) + d \prod_{j=1}^n (s - a_j) \quad (6.27)$$

$$= \left( \sum_{i=1}^n \tilde{c}_i \prod_{j=1, j \neq i}^n (s - a_j) + \prod_{j=1}^n (s - a_j) \right) \frac{\mathbf{n}(s)}{\mathbf{d}(s)}. \quad (6.28)$$

Evaluating this expression at  $s = a_i$  gives

$$c_i \prod_{j=1, j \neq i}^n (a_i - a_j) = \tilde{c}_i \prod_{j=1, j \neq i}^n (a_i - a_j) \frac{\mathbf{n}(a_i)}{\mathbf{d}(a_i)}. \quad (6.29)$$

Therefore,

$$\frac{c_i}{\tilde{c}_i} = \frac{\mathbf{n}(a_i)}{\mathbf{d}(a_i)}, \quad (6.30)$$

so  $\tilde{c}_i = \frac{c_i \mathbf{d}(a_i)}{\mathbf{n}(a_i)}$ . This shows that indeed, for noise-free measurements, at convergence, when  $\tilde{c}_i = 0$ , then  $a_i$  are precisely the system poles, since they are the roots of the denominator  $\mathbf{d}(s) = 0$ .

For noisy measurements, the solution  $\mathbf{x}$  is found from the least squares problem, which can be solved, for instance, via the normal equations:  $\mathbf{A}^* \mathbf{A} \mathbf{x} = \mathbf{A}^* \mathbf{b}$ . In this case,  $\tilde{c}_i$  will be a rational expression involving all  $a_i$ . Since vector fitting converges when  $\tilde{\mathbf{c}} = \mathbf{0}$ , this implies that the convergence points are the roots of the coupled multivariable equations  $\tilde{c}_1(a_1, \dots, a_n) = 0, \dots, \tilde{c}_n(a_1, \dots, a_n) = 0$ . ■

## 6.4 Least squares solutions

The least squares solution of the linear system in Eq. (6.4) can be found by using the normal equations, the QR decomposition or the SVD decomposition.

1. With the *normal equations*, the solution is given as  $\mathbf{x} = (\mathbf{A}^* \mathbf{A})^{-1} \mathbf{A}^* \mathbf{b}$ .
2. On the other hand, we can write  $\mathbf{A}$  as  $\mathbf{QR}$ , where  $\mathbf{Q}$  is unitary ( $\mathbf{Q} \mathbf{Q}^* = \mathbf{Q}^* \mathbf{Q} = \mathbf{I}$ ) and  $\mathbf{R}$  is upper triangular. The solution is then given as  $\mathbf{R} \mathbf{x} = \mathbf{Q}^* \mathbf{b}$ , which can be solved by back substitution.
3. In terms of the *singular value decomposition*,  $\mathbf{A}$  is written as  $\mathbf{U} \mathbf{Q} \mathbf{V}^*$  and, consequently, the solution is  $\mathbf{x} = \mathbf{V} \mathbf{\Sigma}^+ \mathbf{U}^* \mathbf{b}$ , where the pseudoinverse  $\mathbf{\Sigma}^+$  of  $\mathbf{\Sigma}$  is obtained by inverting its non-zero diagonal elements.

These three approaches yield the same result in case the matrix  $\mathbf{A}$  has full column rank. In case  $\mathbf{A}$  has a right nullspace, the solution  $\mathbf{x}$  is not unique, so each of the above methods will return a different solution. Mathematically, the matrix  $\mathbf{A}$  involved in the least problem is full rank if the number of starting poles is chosen less or equal to the order of the underlying system. If the number of starting poles is  $n$  and the order of the underlying system is  $m$  and  $n > m$ , then the number of vectors in the nullspace of  $\mathbf{A}$  is precisely  $n - m$  (see Example 6.7.2.1). In practice, a bad choice of starting poles will lead to more vectors in the nullspace, as shown in Fig.6.1 (which is taken from [18]). The underlying system is of order 18 with poles coming in complex conjugate pairs. When modeling it using 20 complex conjugate starting poles, there are 2 singular values which are below machine precision (dotted line 1) in Fig.6.1). When using 20 real starting poles instead (dotted line 2) in the figure), there are more singular values which are small, indicating that this is a bad choice of starting poles. Dotted lines 3) and 4) show the singular values of  $\mathbf{A}$  obtained when modeling a system with real starting poles using complex and real starting poles, respectively. In this case, more singular values are below machine precision than expected.

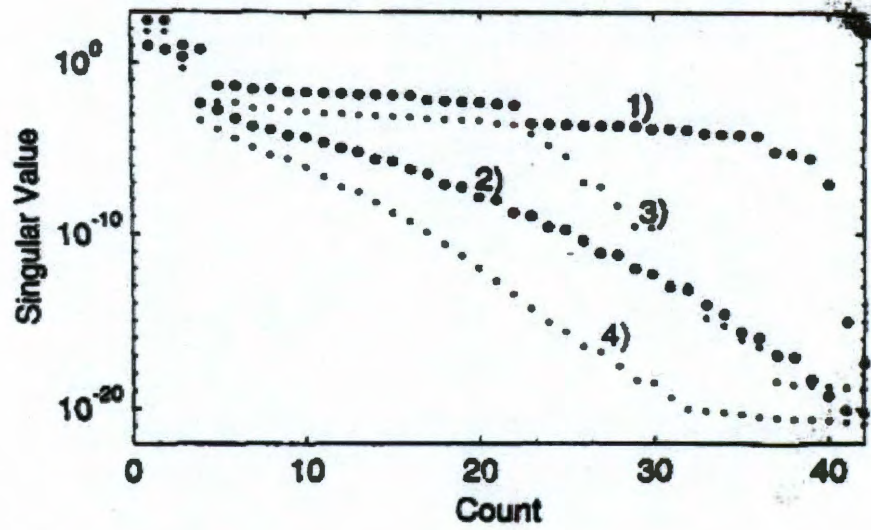


Figure 6.1: Singular values of A (from [18])

## 6.5 Convergence of the pole relocation iteration in vector fitting

We start our analysis from an order 2 strictly proper underlying system, of the form

$$H(s) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2}. \quad (6.31)$$

We model using two starting poles,  $a_1, a_2$ , so the expression we obtain is

$$f(s) = \frac{\frac{c_1}{s-a_1} + \frac{c_2}{s-a_2}}{\frac{\tilde{c}_1}{s-a_1} + \frac{\tilde{c}_2}{s-a_2} + 1} \quad (6.32)$$

$$= \frac{s(c_1 + c_2) - c_1 a_2 - c_2 a_1}{s^2 + s(\tilde{c}_1 + \tilde{c}_2 - a_1 - a_2) + (a_1 a_2 - \tilde{c}_1 a_2 - \tilde{c}_2 a_1)}. \quad (6.33)$$

The poles at the new VF iteration are given by the 2 roots of the polynomial in the denominator

$$s^2 + s(\tilde{c}_1 + \tilde{c}_2 - a_1 - a_2) + (a_1 a_2 - \tilde{c}_1 a_2 - \tilde{c}_2 a_1) = 0. \quad (6.34)$$

Eq. (6.31) contains 4 unknowns, namely  $r_1, r_2, p_1, p_2$ , so a minimum of 4 conditions are needed to determine them. Therefore, at least 4 samples and their corresponding measurements are needed. In case only 4 measurements are provided, Eq. (6.4) leads to the solution  $\mathbf{A}^{-1}\mathbf{b}$ , since the matrix  $\mathbf{A}$  is square.

Suppose we have 5 measurements available of  $H(s)$  at  $s_1, s_2, s_3, s_4$  and  $s_5$ , namely  $H_1, H_2, H_3, H_4$  and  $H_5$ . The resulting linear system which is to be solved is overdetermined:  $\mathbf{A}(s_i, 5; a_i, 2)\mathbf{x}(c_i, 2; \tilde{c}_i, 2) = \mathbf{b}$ , so the solution  $\mathbf{x}$  is found via least squares. The poles at the next iteration are given by the solutions of the  $2^{nd}$  order polynomial in Eq. (6.34). When convergence is reached,  $\tilde{c}_1 = \tilde{c}_2 = 0$ . Using the normal equations, the least squares solution is given as  $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ . It turns out that

$$\tilde{c}_1 = \frac{a_1^4(\gamma_1 a_2^2 + \gamma_2 a_2 + \gamma_3) + a_1^3(\gamma_4 a_2^2 + \gamma_5 a_2 + \gamma_6) + \dots}{a_1^3(\delta_1 a_2^2 + \delta_2 a_2 + \delta_3) + a_1^2(\delta_4 a_2^3 + \delta_5 a_2 + \delta_6) + \dots} = \frac{f_1(a_1, a_2)}{g(a_1, a_2)}, \quad (6.35)$$

$$\tilde{c}_2 = -\frac{a_2^4(\gamma_1 a_1^2 + \gamma_2 a_1 + \gamma_3) + a_2^3(\gamma_4 a_1^2 + \gamma_5 a_1 + \gamma_6) + \dots}{a_1^3(\delta_1 a_2^2 + \delta_2 a_2 + \delta_3) + a_1^2(\delta_4 a_2^3 + \delta_5 a_2 + \delta_6) + \dots} = \frac{f_2(a_1, a_2)}{g(a_1, a_2)}, \quad (6.36)$$

where ... stand for the rest of the terms in the expressions, which contain powers of  $a_1$  and  $a_2$  lower than the ones listed above and  $\delta_i, \gamma_i$  are complicated expressions in terms of  $s_1, s_2, s_3, s_4, s_5$ , and the corresponding measurements  $H_1, H_2, H_3, H_4, H_5$ . We want  $\tilde{c}_1$  and  $\tilde{c}_2$  to be 0 when convergence is reached, so we are looking for the roots of the coupled multivariate polynomials in the numerator:  $f_1(a_1, a_2) = 0$  and  $f_2(a_1, a_2) = 0$ .

**Remark 6.5.1.** *A sequence defined as a fixed point iteration*

$$a^{(k+1)} = h(a^{(k)}) = a^{(k)} - f(a^{(k)})$$

converges to the fixed point  $a$  (which satisfies  $f(a) = 0$ ) if  $|h'(a)| < 1$ . In fact, based on the magnitude of  $h'(a)$ , fixed points can be categorized as attractive, repelling, or indifferent if  $|h'(a)| < 1$ ,  $|h'(a)| > 1$  or  $|h'(a)| = 1$ , respectively. This property

also holds when the iterate is  $\mathbf{a}^{(k)} \in \mathbb{C}^n$  (for  $n$  starting poles), in which case  $h'$  is the Jacobian matrix, while the absolute value is substituted by the spectral radius (the spectral radius of a matrix  $\mathbf{M}$  is defined as the largest eigenvalue of  $\mathbf{M}$  in absolute value). Clearly, if all fixed points are repelling, the iteration will not find any of these solutions and, consequently, will not converge. Also, if there are several attractive fixed points, the iteration will converge to a different solution depending on the location of the initial guess. Ideally, the iteration should have only one attractive fixed point so that it converges to the same solution, irrespective of the initial guess.

One can incorporate a Newton step into the vector fitting iteration to improve the convergence properties of vector fitting in terms of the number of iterations needed to reach convergence, but also, to ensure that the iteration will converge, given a sufficiently good initial guess. In this case, the iteration is of the form

$$\mathbf{a}^{(k+1)} = \mathbf{a}^{(k)} - \mathbf{J}_{\tilde{\mathbf{c}}(\mathbf{a}^{(k)})}^{-1} \tilde{\mathbf{c}}(\mathbf{a}^{(k)}), \quad (6.37)$$

where  $\mathbf{J}_{\tilde{\mathbf{c}}(\mathbf{a}^{(k)})}$  is the Jacobian matrix. The element on its  $i^{\text{th}}$  row and  $j^{\text{th}}$  column defined as  $\frac{\partial \tilde{c}_i(\mathbf{a}^{(k)})}{\partial a_j}$ .

**Example 6.5.1.** We consider the underlying transfer function  $H(s) = \frac{2}{s+5} + \frac{3}{s+10}$  and samples at  $s = 1, s = 2, s = 3, s = 4$  and  $s = 5$ . We corrupt them with additive noise generated using the following line of code in Matlab:

```
H.*(10-3*randn(size(H))),
```

so the added noise is  $10^{-3}$  relative to the magnitude of each entry. The coupled multivariate polynomials that VF is trying to find the roots of are

$$\begin{aligned}
& a_1^4 a_2^2 - 4.73 a_1^4 a_2 + 6.12 a_1^4 - 0.59 a_1^3 a_2^2 + 5.05 a_1^3 a_2 + 10.42 a_1^3 - 31.54 a_1^2 a_2^2 + 148.52 a_1^2 a_2 \\
& - 188.99 a_1^2 + 110.44 a_1 a_2^2 - 589.56 a_1 a_2 + 883.28 a_1 - 115.46 a_2^2 + 706.87 a_2 - 1231.72 = 0 \\
& a_1^2 a_2^4 - 0.59 a_1^2 a_2^3 - 31.54 a_1^2 a_2^2 + 110.44 a_1^2 a_2 - 115.46 a_1^2 - 4.73 a_1 a_2^4 + 5.05 a_1 a_2^3 + 148.52 a_1 a_2^2 \\
& - 589.56 a_1 a_2 + 706.87 a_1 + 6.12 a_2^4 - 10.42 a_2^3 - 188.99 a_2^2 + 883.28 a_2 - 1231.72 = 0.
\end{aligned} \tag{6.38}$$

The two expressions are symmetric in the variables  $a_1$  and  $a_2$ . The roots are listed in Table 6.1, together with the spectral radius of the Jacobian matrix evaluated at each root. Clearly, there are no attractive solutions, since the spectral radius of the Jacobian for all fixed points is larger than 1 (in fact, for the pair (3.25, -6.79), the value is slightly above 1). The results in the last 4 rows make sense since having the two poles equal to each other leads to a Jacobian matrix with 2 linearly dependent columns. We predict that the VF iteration will not converge.

$a_1$	$a_2$	$\ J\ $
3.25	-6.79	1.0004
-6.79	3.25	1.0004
-6.76 ± .26i	2.19 ± 1.14i	2.99
2.19 ± 1.14i	-6.76 ± .26i	2.99
2.22 ± .49i	5.71 ∓ 2.35i	7.26
5.71 ∓ 2.35i	2.22 ± .49i	7.26e+2
3.97 ± .27i	1.98 ± .61i	1.26e+3
1.98 ± .61i	3.97 ± .27i	1.26e+3
3.25	3.25	2.77e+14
2.91 ± .9i	2.91 ± .9i	4.56e+14
-6.78	-6.78	1.19e+16
1.51 ± 1.93i	1.51 ± 1.93i	1.22e+16

Table 6.1: Convergence points and corresponding spectral radius of the Jacobian evaluated at those points

Fig. 6.2(a) and 6.2(b) show the errors at each iteration for the vector fitting algorithm. The iteration process does not converge, but rather oscillates between -6.77



and  $-6.79$  for the first pole and  $3.15$  and  $3.36$  for the second pole. Note that the solution  $(-6.79, 3.25)$ , which is barely repelling, is close to these values found after 100 steps of vector fitting. On the other hand, Fig. 6.2(c) and 6.2(d) show the errors at each iteration when the Newton step is incorporated into the vector fitting iteration, which indicate that the solution is found. Last, Fig. 6.2(e) and 6.2(f) show the iterates at each step when applying VARPRO with the trust-region approach in the nonlinear least squares solver. Using the *trust-region-reflective* algorithm in Matlab's `lsqnonlin` function yields the following output "Optimization stopped because the relative sum of squares ( $r$ ) is changing by less than options.TolFun =  $2.220446e-016$ " after 33 steps and the answer returned is  $-4.31$  and  $-3086.51$ . Generally, VARPRO yields poles which are closer to the location of the underlying ones. The reason for the poor performance for this example is the low number of measurements considered (only 5). All methods used an initial set of poles located at  $-15$  and  $-20$ , respectively.

**Remark 6.5.2.** If the solution contains poles in the right-half plane, by flipping the unstable poles in the left-half plane at each step, the vector fitting iteration will obviously not find the true solution and will, consequently, not converge.

We turn now to analyze modeling a strictly proper underlying system of order 1:

$$H(s) = \frac{r}{s - p}. \quad (6.39)$$

We use one starting pole  $a_1$ , so the expression we obtain is

$$f(s) = \frac{\frac{c_1}{s - a_1}}{\frac{\tilde{c}_1}{s - a_1} + 1} = \frac{c_1}{s - (a_1 - \tilde{c}_1)}. \quad (6.40)$$

At the next iteration, the pole is given by the zero of the polynomial in the denominator, namely  $a_1 - \tilde{c}_1$ , where  $\tilde{c}_1$  is found by solving the least squares problem in Eq.

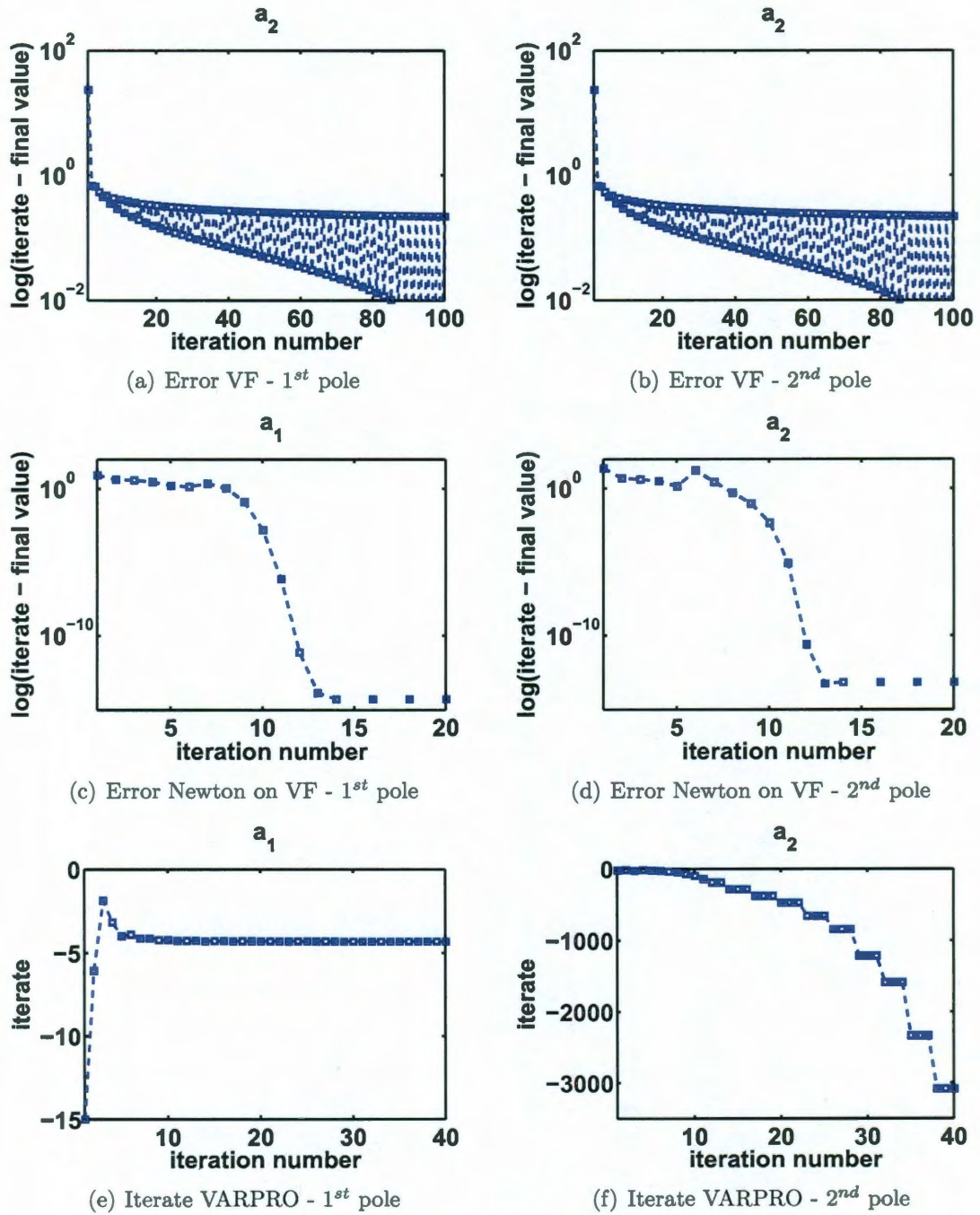


Figure 6.2: Errors or iterates for the various methods considered

(6.4), yielding the following iteration

$$a_1^{(k)} = a_1^{(k)} - \tilde{c}_1^{(k)}.$$

**Example 6.5.2.** Assume that we are provided with 3 measurements at  $s = 2$ ,  $s = 6$  and  $s = 7$  with the following values  $H_1 = 8$ ,  $H_2 = 7$ ,  $H_3 = 1$ . The least squares problem becomes:

$$\underbrace{\begin{bmatrix} \frac{1}{2-a} & -\frac{8}{2-a} \\ \frac{1}{6-a} & -\frac{7}{6-a} \\ \frac{1}{7-a} & -\frac{1}{7-a} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_1 \\ \tilde{c}_1 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 8 \\ 7 \\ 1 \end{bmatrix}}_{\mathbf{b}} \quad (6.41)$$

The solution  $\mathbf{x}$ , obtained via the normal equations, leads to the following VF iteration (we change the notation of  $a_1^{(k)}$  to  $a_k$  to avoid double superscripts):

$$a_{k+1} = a_k - \tilde{c}_1 = a_k - \frac{86a_k^3 - 993a_k^2 + 3189a_k - 1834}{86a_k^2 - 746a_k + 1957} \quad (6.42)$$

$$= a_k - \frac{f(a_k)}{g(a_k)} = h(a_k), \quad (6.43)$$

so the polynomial VF is trying to find the roots of  $86a^3 - 993a^2 + 3189a - 1834$ . The 3 roots computed in Matlab using the command `roots` are 5.67, 5.14 and .73. When computing  $h'(a)$ , to check which of the fixed points are attractive, repelling, or indifferent, we obtain:  $h'(5.67) = .54$ ,  $h'(5.14) = 1.5$ ,  $h'(.73) = -.28$ . In this case, there are two roots which are attractive, namely 5.67 and .73, so VF may converge to either of the two solutions depending on the location of the initial guess, situation which is not desirable. Fig. 6.3 shows the “basins of attraction” associated to the VF iteration in this example. The two colors correspond to areas in the complex plane which, when used as an initial guess for the VF iteration, converge to one of the two attractive solutions; no initial guess will lead to convergence to the repelling solution, hence only the two areas. The three points on the real axis are the 3 roots (the black

and the blue points are the attractive solutions, while the red one is the repelling root). VARPRO converges to a stable pole located at  $-2.32$  (while VF converges to  $.73$ ) when

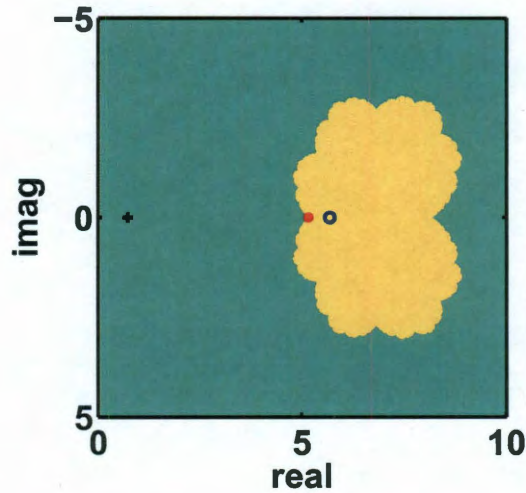


Figure 6.3: “Basins of attraction” for the VF iteration

using an initial guess located at  $-100$ . When using a starting pole located at  $5.9$ , VF converges to  $5.67$ , while VARPRO yields  $6.08$ , which is unstable (due to the fact that the initial guess was located in the right-half plane).

## 6.6 Incorporating the Newton step in the vector fitting iteration

Incorporating the Newton step in the vector fitting iteration offers several advantages. The most important advantage is that, even though the vector fitting iteration may not converge for a certain data set, by making use of the Newton step, convergence is guaranteed for an initial guess located close to the solution. Also, the Newton step may improve the convergence properties in terms of the number of iterations needed for convergence, as the vector fitting iteration by itself may converge slower than when the Newton step is included. In the sequel, details about the addition of

the Newton step to the VF iteration are provided.

As mentioned previously, at convergence,  $\tilde{\mathbf{c}} = \mathbf{0}$ , so vector fitting is essentially solving for the zeros of a set of multivariate functions  $\tilde{\mathbf{c}}(\mathbf{a})$ , where  $\tilde{\mathbf{c}}$  are found from the least squares system  $\mathbf{A}(s_i, N; a_i, n)\mathbf{x}(c_i, n; d; \tilde{c}_i, n) = \mathbf{b}$  detailed in Eq. (6.4). This problem can also be solved by the *Newton iteration*:

$$\mathbf{a}^{(k+1)} \leftarrow \mathbf{a}^{(k)} - \mathbf{J}_{\tilde{\mathbf{c}}}^{-1}\tilde{\mathbf{c}}(\mathbf{a}^{(k)}), \quad (6.44)$$

where  $\mathbf{J}_{\tilde{\mathbf{c}}}$  is the Jacobian matrix obtained by taking partial derivatives of the vector  $\tilde{\mathbf{c}}$  with respect to the vector of unknowns  $\mathbf{a}$ . Its  $i^{\text{th}}$  column is  $\mathbf{J}_{\tilde{\mathbf{c}}}(:, i) = \frac{\partial \tilde{\mathbf{c}}}{\partial a_i}$ . To ensure convergence, a steepest descent method can be employed, in which case the iteration is  $\mathbf{a}^{(k+1)} \leftarrow \mathbf{a}^{(k)} - t_k \mathbf{J}_{\tilde{\mathbf{c}}}^{-1}\tilde{\mathbf{c}}(\mathbf{a}^{(k)})$ , where  $t_k$  is 1, if the residue  $\tilde{\mathbf{c}}(\mathbf{a}^{(k)})$  at the current step is smaller than the residue  $\tilde{\mathbf{c}}(\mathbf{a}^{(k-1)})$  at the previous step, otherwise the step size  $t_k$  is chosen to ensure that the residue always decreases.

### 6.6.0.1 Recursive Implementation

Computing the partial derivative with respect to  $a_i$  of the least squares problem  $\mathbf{A}(s_i, N; a_i, n)\mathbf{x}(c_i, n; d; \tilde{c}_i, n) = \mathbf{b}$ , we obtain

$$\frac{\partial \mathbf{A}(s_i, N; a_i, n)}{\partial a_i} \mathbf{x}(c_i, n; d; \tilde{c}_i, n) + \mathbf{A}(s_i, N; a_i, n) \frac{\partial \mathbf{x}(c_i, n; d; \tilde{c}_i, n)}{\partial a_i} = \mathbf{b}.$$

It is easy to see that  $\frac{\partial \mathbf{A}}{\partial a_i}$  is given by

$$\frac{\partial \mathbf{A}(s_i, N; a_i, n)}{\partial a_i} = \begin{bmatrix} \frac{\partial \Phi}{\partial a_i} & \mathbf{0} & -\text{diag}(\mathbf{b}) \frac{\partial \Phi}{\partial a_i} \end{bmatrix}, \quad \frac{\partial \Phi}{\partial a_i} = \begin{bmatrix} 0 & \dots & \frac{1}{(s_1 - a_i)^2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \frac{1}{(s_N - a_i)^2} & \dots & 0 \end{bmatrix}, \quad (6.45)$$

so

$$\mathbf{A}(s_i, N; a_i, n) \frac{\partial \mathbf{x}(c_i, n; d; \tilde{c}_i, n)}{\partial a_i} = \mathbf{b} - \frac{\partial \mathbf{A}(s_i, N; a_i, n)}{\partial a_i} \mathbf{x}(s_i, N; a_i, n), \quad (6.46)$$

leading to Algorithm 3, included in Appendix B.3.

### 6.6.0.2 Implementation with normal equations

Using the *normal equations* to solve the least squares problem, we have  $(\mathbf{A}^* \mathbf{A}) \mathbf{x} = \mathbf{A}^* \mathbf{b}$ .

Taking partial derivatives with respect to  $a_i$ , we obtain

$$\frac{\partial (\mathbf{A}^* \mathbf{A})}{\partial a_i} \mathbf{x} + (\mathbf{A}^* \mathbf{A}) \frac{\partial \mathbf{x}}{\partial a_i} = \left( \frac{\partial \mathbf{A}}{\partial a_i} \right)^* \mathbf{b}, \quad (6.47)$$

so, solving for  $\frac{\partial \mathbf{x}}{\partial a_i}$ , we have that

$$(\mathbf{A}^* \mathbf{A}) \frac{\partial \mathbf{x}}{\partial a_i} = \left( \frac{\partial \mathbf{A}}{\partial a_i} \right)^* \mathbf{b} - \left[ \left( \frac{\partial \mathbf{A}}{\partial a_i} \right)^* \mathbf{A} + \mathbf{A}^* \frac{\partial \mathbf{A}}{\partial a_i} \right] \mathbf{x}, \quad (6.48)$$

where  $\frac{\partial \mathbf{A}}{\partial a_i}$  is the same as in Eq.(6.46).

This development yields Algorithm 4 in Appendix B.4.

### 6.6.0.3 Implementation employing the pseudoinverse

Using the pseudoinverse to express the solution  $\mathbf{x}$ , we have  $\mathbf{x} = \mathbf{A}^+ \mathbf{b}$ . The pseudoinverse  $\mathbf{A}^+$  can be computed in Matlab using the command `pinv`, which uses an implementation via SVD. Taking partial derivatives of this expression, we get

$$\frac{\partial \mathbf{x}}{\partial a_i} = \left( \frac{\partial \mathbf{A}^+}{\partial a_i} \right) \mathbf{b}. \quad (6.49)$$

The partial derivative of the pseudoinverse is computed as [29]

$$\left( \frac{\partial \mathbf{A}^+}{\partial a_i} \right) = -\mathbf{A}^+ \left( \frac{\partial \mathbf{A}}{\partial a_i} \right) \mathbf{A}^+ + \mathbf{A}^+ (\mathbf{A}^+)^* \left( \frac{\partial \mathbf{A}}{\partial a_i} \right)^* \mathbf{P}_\mathbf{A}^\perp + \mathbf{A} \mathbf{P}^\perp \left( \frac{\partial \mathbf{A}}{\partial a_i} \right)^* (\mathbf{A}^+)^* \mathbf{A}^+, \quad (6.50)$$

where  $\mathbf{P}_\mathbf{A}^\perp = \mathbf{I} - \mathbf{A} \mathbf{A}^+$  and  ${}_\mathbf{A} \mathbf{P}^\perp = \mathbf{I} - \mathbf{A}^+ \mathbf{A}$ . This development yields Algorithm 5 in Appendix B.5.

## 6.7 Proposed reformulation of vector fitting

We aim at finding the transfer function of the underlying system, which is a rational expression, equal to a ratio of two polynomials:

$$f(s) = \frac{N(s)}{D(s)} = \frac{a_0 + a_1s + a_2s^2 + \dots + a_ns^n}{b_0 + b_1s + b_2s^2 + \dots + b_ns^n}. \quad (6.51)$$

One can express polynomials in various bases. Above, the expressions  $N(s)$  and  $D(s)$  are in the monomial basis, which is known to be ill-conditioned for a high degree  $n$  (as in real applications). A better suited basis is that of Lagrange polynomials:

$$q_i(s) = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{s - \alpha_j}{\alpha_i - \alpha_j},$$

for  $\alpha_j$  chosen a-priori. A polynomial  $N(s)$  can be expressed in the Lagrange basis as

$$N(s) = \sum_{i=0}^n \beta_i q_i(s).$$

It is easy to see that the following interpolation conditions hold:  $N(\alpha_i) = \beta_i$ ,  $i = 0, \dots, n$ . Changing notation, we can express the transfer function as

$$f(s) = \frac{N(s)}{D(s)} = \frac{c_0 \hat{q}_0(s) + c_1 \hat{q}_1(s) + c_2 \hat{q}_2(s) + \dots + c_n \hat{q}_n(s)}{\tilde{c}_0 \hat{q}_0(s) + \tilde{c}_1 \hat{q}_1(s) + \tilde{c}_2 \hat{q}_2(s) + \dots + \tilde{c}_n \hat{q}_n(s)}. \quad (6.52)$$

with the numerator and denominator in the Lagrange basis, where

$$\hat{q}_i(s) = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} (s - a_j)$$

and the constant factor  $\prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{1}{(a_i - a_j)}$  is incorporated in  $c_i$  and  $\tilde{c}_i$ , respectively. One

can divide the numerator and denominator by  $\prod_{0 \leq j \leq n} (s - a_j)$  to obtain:

$$f(s) = \frac{N(s)}{D(s)} = \frac{\frac{c_0}{s-a_0} + \frac{c_1}{s-a_1} + \frac{c_2}{s-a_2} + \dots + \frac{c_n}{s-a_n}}{\frac{\tilde{c}_0}{s-a_0} + \frac{\tilde{c}_1}{s-a_1} + \frac{\tilde{c}_2}{s-a_2} + \dots + \frac{\tilde{c}_n}{s-a_n}}. \quad (6.53)$$

**Remark 6.7.1.** *Note that the above is an expression of the transfer function in the Lagrange basis, rather than the rational basis (or a partial fraction), since polynomials cannot be expressed in a rational basis.*

The expression in Eq. (6.53) is similar to the one in Eq. (6.2), repeated below for convenience:

$$\begin{bmatrix} \mathbf{n}(s) \\ \mathbf{d}(s) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n \frac{c_i}{s-a_i} + d \\ \sum_{i=1}^n \frac{\tilde{c}_i}{s-a_i} + 1 \end{bmatrix}.$$

Suppose the underlying transfer function is of order  $n$ , as expressed in Eq. (6.51), so it is clear from Eq. (6.53), that the number of required starting poles  $a_i$  is  $n+1$ . The following remark is in place.

**Remark 6.7.2.** *The number of starting poles in the original vector fitting procedure is the same as the order  $n$  of the underlying system. The restriction set on the denominator to approach 1 for high frequencies can be removed by introducing an additional starting pole.*

Next, we discuss the issue of convergence for the proposed reformulation. At iteration  $k$ , the denominator is

$$D(s) = \frac{k_1 \prod_{i=1}^n (s - a_i^{(k+1)})}{k_2 (s - a_0) \prod_{i=1}^n (s - a_i^{(k)})},$$

so when convergence occurs at iteration  $k+1$ , then  $a_i^{(k+1)} = a_i^{(k)}$ , therefore,  $D(s) = \frac{k_1}{k_2 (s - a_0)} = \frac{\tilde{c}_0}{s - a_0}$ . In the proposed reformulation, convergence occurs when  $\tilde{c}_i = 0$ ,  $i = 1, \dots, n$  (like in the original VF formulation).

**Remark 6.7.3.** *Note that interpolation at  $s = a_i$  is exact both in the case of the regular Lagrange expansion, as well as in the one employed by vector fitting (see Eq.(6.30)). This is due to the fact that, when evaluating  $f(s)$  at  $s = a_i$ , the result is  $f(a_i) = \frac{c_i}{\tilde{c}_i}$ .*



Interpolation is exact at  $s = a_i$ , so we investigate the issue of choosing  $a_0, a_1, \dots, a_n$  at the first iteration. To enforce interpolation at DC or high frequencies, one can use the extra starting pole at 0, namely  $a_0 = 0$ , or some high value. As for the other starting poles, currently VF uses a heuristic to select the starting poles, as complex conjugate pairs  $a_i = -\alpha + \beta j$ ,  $a_{i+1} = -\alpha - \beta j$ , with the imaginary part  $\beta$  linearly distributed over the frequency range of interest, and  $\alpha = \frac{\beta}{100}$ . The following remark is in place.

**Remark 6.7.4.** *Since interpolation is exact at the starting poles  $a_i$ , the best choice is to choose them as purely imaginary quantities, namely  $j\omega_i = j \cdot 2\pi f_i$ , where  $f_i$  are some of the frequencies where measurements are provided. This change implies that, when setting up the least squares problem  $\mathbf{A}\mathbf{x} = \mathbf{b}$  in the pole identification stage, one should use the remaining measurements to set up Eq. (6.4).*

Multiplying Eq. (6.53) with the denominator yields:

$$\sum_{i=0}^n \frac{c_i}{s - a_i} - \left( \sum_{i=0}^n \frac{\tilde{c}_i}{s - a_i} \right) f(s) = 0. \quad (6.54)$$

Vector fitting uses the fact that  $f(s)$  interpolates the underlying transfer function at the samples to obtain an overdetermined least squares problem  $\mathbf{A}\mathbf{x} = \mathbf{0}$  where the unknowns are in the solution vector  $\mathbf{x}$  like in Eq. (6.4), except that in this reformulation, the right hand side is  $\mathbf{0}$ :

$$\underbrace{\hat{\mathbf{A}}(s_i, N; a_i, n)}_{\in \mathbb{C}^{N \times (2(n+1))}} \underbrace{\hat{\mathbf{x}}(c_i, n; \tilde{c}_i, n)}_{\in \mathbb{C}^{(2(n+1)) \times 1}} = \underbrace{\mathbf{0}}_{\in \mathbb{C}^{N \times 1}}, \text{ where} \quad (6.55)$$

$$\hat{\mathbf{A}}(s_i, N; a_i, n) = \left[ \hat{\Phi}(s_i, N; a_i, n) - \text{diag}(\mathbf{b}) \hat{\Phi}(s_i, N; a_i, n) \right], \quad (6.56)$$

$$\hat{\Phi}(s_i, N; a_i, n) = \begin{bmatrix} \frac{1}{s_1 - a_0} & \cdots & \frac{1}{s_1 - a_n} \\ \vdots & \ddots & \vdots \\ \frac{1}{s_N - a_0} & \cdots & \frac{1}{s_N - a_n} \end{bmatrix} \quad (6.57)$$

$$\hat{\mathbf{x}}(c_i, n; \tilde{c}_i, n) = \begin{bmatrix} \hat{\mathbf{c}} \\ \hat{\tilde{\mathbf{c}}} \end{bmatrix}, \quad \text{where } \hat{\mathbf{c}} = \begin{bmatrix} c_0 \\ \vdots \\ c_n \end{bmatrix}, \quad \hat{\tilde{\mathbf{c}}} = \begin{bmatrix} \tilde{c}_0 \\ \vdots \\ \tilde{c}_n \end{bmatrix}. \quad (6.58)$$

Eq. (6.55) can be solved by selecting the right singular vector corresponding to the smallest singular value of  $\mathbf{A}$  (in case the matrix  $\mathbf{A}$  does not have a right nullspace, which may occur when noise is added to the measurements, like in the example in Sect. 6.7.2.2).

Once the quantities  $c_i, \tilde{c}_i, i = 0, \dots, n$  are found, we have recovered the transfer function of the original system with the numerator and denominator polynomials expressed in a Lagrange basis and we can write down a realization immediately (similar to Lemma 6.1.1, so the proof is omitted this time):

**Lemma 6.7.1.** [51] *The following is a descriptor realization of  $f(s)$  expressed as in*

Eq. (6.59):  $f(s) = \mathbf{C}\Phi(s)^{-1}\mathbf{B}$ , where

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & \dots & c_n \end{bmatrix}, \quad (6.59)$$

$$\Phi(s) = \begin{bmatrix} (s-a_0) & (a_1-s) & & & \\ (s-a_0) & 0 & (a_2-s) & & \\ \vdots & & \ddots & \ddots & \\ (s-a_0) & & & 0 & (a_n-s) \\ \tilde{c}_0 & \tilde{c}_1 & \tilde{c}_2 & \dots & \tilde{c}_n \end{bmatrix}, \quad (6.60)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{n \times 1} \\ 1 \end{bmatrix}. \quad (6.61)$$

**Remark 6.7.5.** *At this point, in case of noise-free measurements, we have recovered the original system. Otherwise, for noisy measurements, an approximate is obtained. For a better approximation, one can employ an iteration, like the pole relocation iteration. Inspired by the ideas of VF, one can use the  $n$  poles of the present model as the new starting poles  $a_i$ , together with the additional starting pole  $a_0$ .*

**Remark 6.7.6.** *If the starting poles are selected as some of the measurements, the unknowns  $\tilde{c}_i$  and  $c_i$  are related by*

$$\underbrace{f(a_i)}_{=H(s_i)=H_i} = \frac{c_i}{\tilde{c}_i} \Rightarrow c_i = H_i \tilde{c}_i, \quad (6.62)$$

so the least squares  $\mathbf{Ax} = \mathbf{0}$  can be set up with only half of the unknowns.

**Remark 6.7.7. Connection to the Loewner framework.** *Select  $a_i$ ,  $i = 0, \dots, n$  as some of the measurements, denoted as  $\lambda_i$ . Thus, as discussed previously,  $\mathbf{H}(\lambda_i)$ , which we denote as  $\mathbf{w}_i$  can be expressed as  $\mathbf{w}_i = \frac{c_i}{\tilde{c}_i}$ . Since the Lagrange basis formulation of  $f(s)$  can be rewritten as  $f(s) = \frac{\sum_{i=0}^n \frac{c_i}{s-\lambda_i}}{\sum_{i=0}^n \frac{\tilde{c}_i}{s-\lambda_i}}$ , then  $f(s)$  satisfies the equation*

$$\sum_{i=0}^n \tilde{c}_i \frac{f(s) - \mathbf{w}_i}{s - \lambda_i} = 0, \quad \tilde{c}_i \neq 0. \quad (6.63)$$

To determine  $\tilde{c}_i$  (and consequently,  $c_i$ , from  $c_i = \mathbf{w}_i \tilde{c}_i$ ), consider the rest of the interpolation conditions: we write Eq. (6.63) for the rest of the  $N - (n + 1)$  measurements:

$$\mathbf{H}(\mu_j) = \mathbf{v}_j, \quad j = 0, \dots, N - (n + 1) - 1, \quad (6.64)$$

where  $\mu_j$  satisfy  $\mu_i \neq \mu_j$ ,  $i \neq j$ ,  $\lambda_i \neq \mu_j$ ,  $\forall i, j$ . Substituting Eq. (6.64) in Eq. (6.63) we obtain the following condition for  $\tilde{c}_i$ :  $\mathbb{L}\tilde{\mathbf{c}} = \mathbf{0}$ , where

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{v}_0 - \mathbf{w}_0}{\mu_0 - \lambda_0} & \dots & \frac{\mathbf{v}_0 - \mathbf{w}_n}{\mu_0 - \lambda_n} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_m - \mathbf{w}_0}{\mu_m - \lambda_0} & \dots & \frac{\mathbf{v}_m - \mathbf{w}_n}{\mu_m - \lambda_n} \end{bmatrix} \in \mathbb{C}^{(m+1) \times (n+1)}, \quad \tilde{\mathbf{c}} = \begin{bmatrix} \tilde{c}_0 \\ \vdots \\ \tilde{c}_n \end{bmatrix} \in \mathbb{C}^{n+1}, \quad (6.65)$$

with  $m = N - (n + 1) - 1$ . The matrix  $\mathbb{L}$  is the Loewner matrix [52, 53, 44, 2] associated with the row array  $(\mu_j, \mathbf{v}_j)$ , and the column array  $(\lambda_i, \mathbf{w}_i)$ . Thus  $f(s)$  is determined if  $\tilde{\mathbf{c}}$  is in the right kernel of  $\mathbb{L}$  (and  $\tilde{c}_i \neq 0$ ). A key fact is that the degree of the underlying rational function, defined as the maximum between the degrees of the polynomials in the numerator and denominator, is equal to the rank of  $\mathbb{L}$ .

**Lemma 6.7.2.** *As a consequence of the fact that the matrix  $\mathbf{A}$  in Eq. (6.56) is a variant of the Loewner matrix, its rank is also related to the McMillan degree of the underlying system. Its nullspace contains  $k$  vectors if the number of starting poles is  $n + k$ , where  $n$  is the McMillan degree of the underlying system.*

### 6.7.1 Lagrange basis in VF

Vector fitting uses the rational basis instead of the Lagrange basis, so the solution obtained is different. Eq. (6.5) - (6.7) using the Lagrange basis are

$$\begin{aligned}
 \mathbf{A} &= \underbrace{\begin{bmatrix} \hat{q}_1(s_1) & \dots & \hat{q}_n(s_1) & \prod_{i=1}^n (s_1 - a_i) & -H_1 \hat{q}_1(s_1) & \dots & -H_1 \hat{q}_n(s_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{q}_1(s_N) & \dots & \hat{q}_n(s_N) & \prod_{i=1}^n (s_N - a_i) & -H_1 \hat{q}_1(s_N) & \dots & -H_1 \hat{q}_n(s_N) \end{bmatrix}}_{\mathbf{DA}_{VF}}, \\
 \mathbf{x} &= \begin{bmatrix} c_1 & \dots & c_n & d & \tilde{c}_1 & \dots & \tilde{c}_n \end{bmatrix}^T, \\
 \mathbf{b} &= \underbrace{\begin{bmatrix} H_1 \prod_{i=1}^n (s_1 - a_i) & \dots & H_N \prod_{i=1}^n (s_N - a_i) \end{bmatrix}^T}_{\mathbf{Db}_{VF}}, \tag{6.66}
 \end{aligned}$$

where  $\mathbf{D} = \text{diag}(\prod_{i=1}^n (s_1 - a_i), \dots, \prod_{i=1}^n (s_N - a_i))$  and  $\mathbf{A}_{VF}$  and  $\mathbf{b}_{VF}$  are the ones in Eq. (6.5) and (6.6). Naturally, if the number of equations is the same as the number of unknowns (that is, the number of initial starting poles  $n$  is selected equal to the number of samples  $N$ ), the solution using the Lagrange basis is the same as the one obtained in VF:

$$\mathbf{x} = (\mathbf{DA}_{VF})^{-1} \mathbf{Db}_{VF} = \mathbf{A}_{VF}^{-1} \mathbf{D}^{-1} \mathbf{Db}_{VF} = \mathbf{x}_{VF}.$$

However, if one needs to solve the linear system as a least squares problem, the solutions obtained via the two approaches will differ. The solution using the normal equations is given as

$$\mathbf{x}_{VF} = (\mathbf{A}_{VF}^T \mathbf{A}_{VF})^{-1} \mathbf{A}_{VF}^T \mathbf{b}_{VF},$$

in the vector fitting case, as opposed to

$$\mathbf{x} = (\mathbf{A}_{VF}^T \mathbf{D}^T \mathbf{D} \mathbf{A}_{VF})^{-1} \mathbf{A}_{VF}^T \mathbf{D}^T \mathbf{D} \mathbf{b}_{VF},$$

in the Lagrange case. Clearly,  $\mathbf{x}_{VF} \neq \mathbf{x}$ .

### 6.7.2 Numerical example

The accuracy of the models is assessed using two error measures:

- the normalized  $\mathcal{H}_\infty$ -norm of the error system, which is the absolute value of the largest deviation between the model and the measurements divided by the absolute value of the largest value of all measurements:

$$\mathcal{H}_\infty \text{ error} = \frac{\max_{i=1\dots k} |f(j\omega_i) - H_i|}{\max_{i=1\dots k} |H_i|},$$

- the normalized  $\mathcal{H}_2$ -norm of the error system, which is the sum of the square of the magnitude of the deviation between the model and the measurements at all samples, divided by the sum of the square of the magnitude of all measurements:

$$\mathcal{H}_2 \text{ error} = \sqrt{\frac{\sum_{i=1}^k |f(j\omega_i) - H_i|^2}{\sum_{i=1}^k |H_i|^2}}.$$

The first error measure evaluates the maximum deviation between the model and the measurements, while the second one evaluates the deviation at all points, proving to be a good estimate of the overall performance of the model.

## 6.7.2.1 Noise-free measurements

Let us consider the following transfer function:

$$H(s) = \frac{2}{s+5} + \frac{30+j40}{s - (-100+j500)} + \frac{30-j40}{s - (-100-j500)} + 0.5,$$

which we sample at  $N_s = 101$  points distributed as  $\mathbf{s}=2*\pi*j*\text{logspace}(0,4,N_s)$ .

We also append the complex conjugates of these data as the information at  $-j\omega_i$ .

We compare the original formulation of VF to the proposed reformulation in terms of the resulting errors. The starting poles were chosen real, for simplicity, more precisely as  $\text{poles}=-2*\pi*\text{logspace}(0,4,n)$ , in all cases, where  $n$ , the approximation order, was chosen 3.

The errors in our proposed reformulation depend on whether we choose the extra starting pole to be 0 or at some large value. We discuss both cases. Tables 6.2 and 6.3 present the  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$  errors at the first 2 iterations, with all the original, as well as the proposed reformulation. The proposed reformulation with an extra pole located at 0 yields errors in the same range as those with the original VF formulation, while those obtained with the extra pole located at  $10^6$  are slightly larger.

Iteration	Original VF	New VF with extra pole at 0	New VF with extra pole at $10^6$
1	3.5399e-012	2.3275e-014	1.1669e-011
2	2.9933e-015	1.5174e-015	8.1688e-013

Table 6.2:  $\mathcal{H}_\infty$  error

Iteration	Original VF	New VF with extra pole at 0	New VF with extra pole at $10^6$
1	1.6864e-012	1.1000e-014	4.8523e-012
2	1.3394e-015	1.1515e-015	3.7602e-013

Table 6.3:  $\mathcal{H}_2$  error

Next, we show on this numerical example that the dimension of the underlying

system can be deduced from the nullspace of matrix  $\mathbf{A}$  (used in solving the least squares problem). Fig 6.4 shows the decay of the normalized singular values of the matrices  $\mathbf{A}$  at the first iteration for all approaches. Note that, while the matrix resulting from Relaxed VF is full rank, the ones resulting from our proposed approach have one vector in the nullspace.

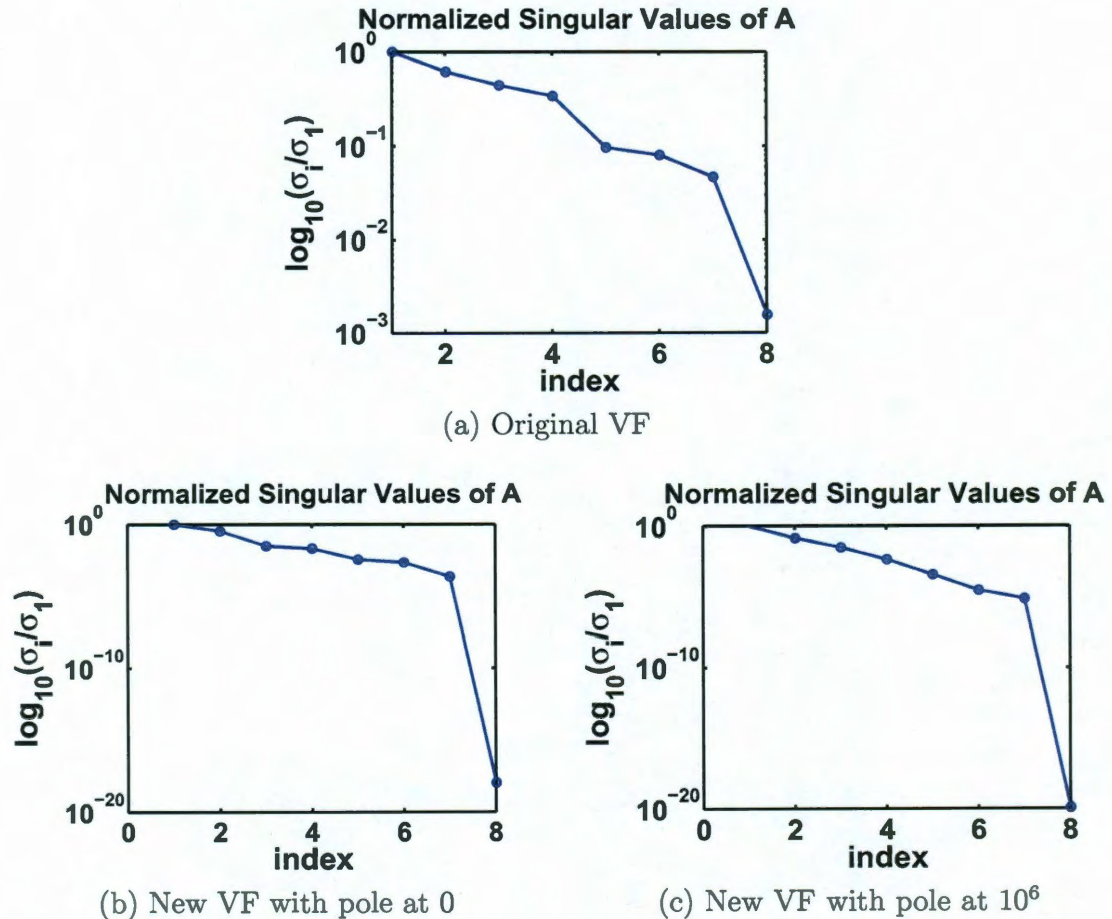


Figure 6.4: Normalized singular values of  $\mathbf{A}$  when the model order is chosen 3

Fig 6.5 shows the decay of the normalized singular values of the matrices  $\mathbf{A}$  at the first 3 iterations for all cases, when considering  $n = 5$  starting poles (i.e., 2 more than actually needed). The matrix  $\mathbf{A}$  resulting from the original VF formulation is no longer full rank, but rather has a nullspace of dimension 2, indicating that the least squares solution to the linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$  is not unique, since one can add



any linear combination of vectors in the null space to the solution  $\mathbf{x}$ . On the other hand, the matrices  $\mathbf{A}$  resulting from our proposed reformulation no longer contain a single vector in the null-space, but rather 3 (i.e., 2 more than needed).

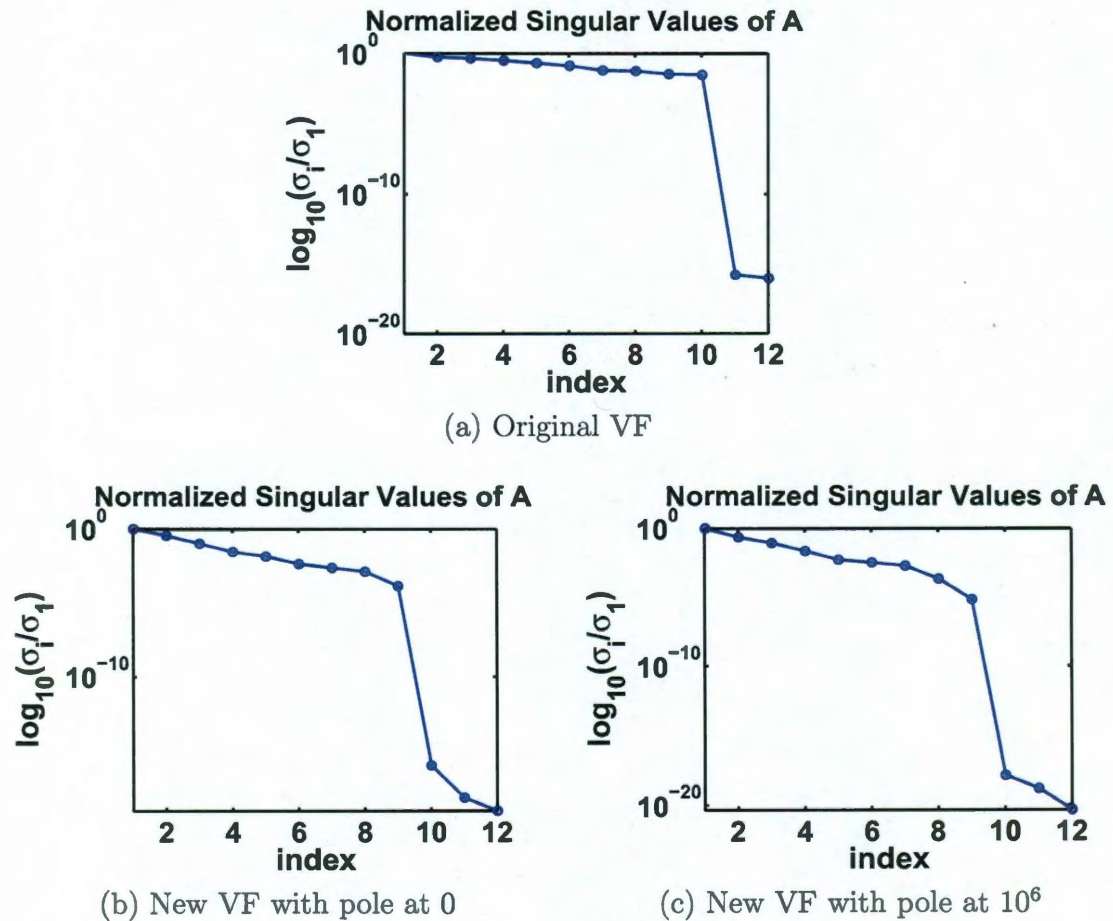


Figure 6.5: Normalized singular values of  $\mathbf{A}$  when the model order is chosen 5

### 6.7.2.2 Noisy measurements

We consider the same transfer function, to which we add noise as

$$N_i = H(s_i) \cdot (10^{-2}(\text{randn} + j * \text{randn})).$$

For comparison, we compute the  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$ -norms of the noise:

- the  $\mathcal{H}_\infty$ -norm of the noise we introduced:  $\mathcal{H}_\infty \text{ norm} = \max_{i=1\dots k} |N_i|$ ,
- the  $\mathcal{H}_2$ -norm of the noise we introduced:  $\mathcal{H}_2 \text{ norm} = \sqrt{\sum_{i=1}^k |N_i|^2}$ .

Recall that at convergence, the vector  $\tilde{\mathbf{c}}$  is  $\mathbf{0}$ , so we compare all methods discussed previously in terms of errors obtained when convergence is achieved. For the original VF formulation, the norm of  $\tilde{\mathbf{c}}$  after 11 iterations is  $8.1556e - 014$  (and will not decrease when increasing the number of iterations). For the proposed reformulation with an extra pole at 0 and  $10^6$ , the errors are  $6.8791e - 016$  and  $1.1286e - 016$  after 20 and 7 iterations, respectively. VARPRO stops after 52 iterations returning the message “lsqnonlin stopped because the problem appears to be locally singular”.

Tables 6.4 presents the  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$  errors obtained with all methods. The smallest  $\mathcal{H}_\infty$  errors are obtained using the proposed method with the extra pole at  $10^6$ , while the smallest  $\mathcal{H}_2$  error is obtained with VARPRO. Still, the errors obtained with original VF, proposed VF with extra pole at  $10^6$  and VARPRO are very close to each other.

For comparison, the  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$  norms of the error vector are  $1.8627e-002$  and  $1.4010e-002$ , respectively, so the errors obtained with the proposed reformulation with the extra pole at 0 are larger than that, but the rest of the methods yield errors slightly below the noise values.

	Original VF	New VF with extra pole at 0	New VF with extra pole at $10^6$	VARPRO
$\mathcal{H}_\infty$ error	1.6985e-002	2.6649e-002	1.6983e-002	1.6988e-002
$\mathcal{H}_2$ error	1.3411e-002	1.6112e-002	1.3411e-002	1.3408e-002

Table 6.4: Errors with the different algorithms

Next, we show the decay of the singular values of the matrix  $\mathbf{A}$  (used in solving the linear system). Fig 6.6 show the decay of the normalized singular values of the matrices  $\mathbf{A}$  at the first iteration for all the methods we are comparing. Note that, while the matrix resulting from Relaxed VF is still full rank, the ones resulting from

our proposed approach should be singular but, due to the added noise, the last singular value is no longer below machine precision.

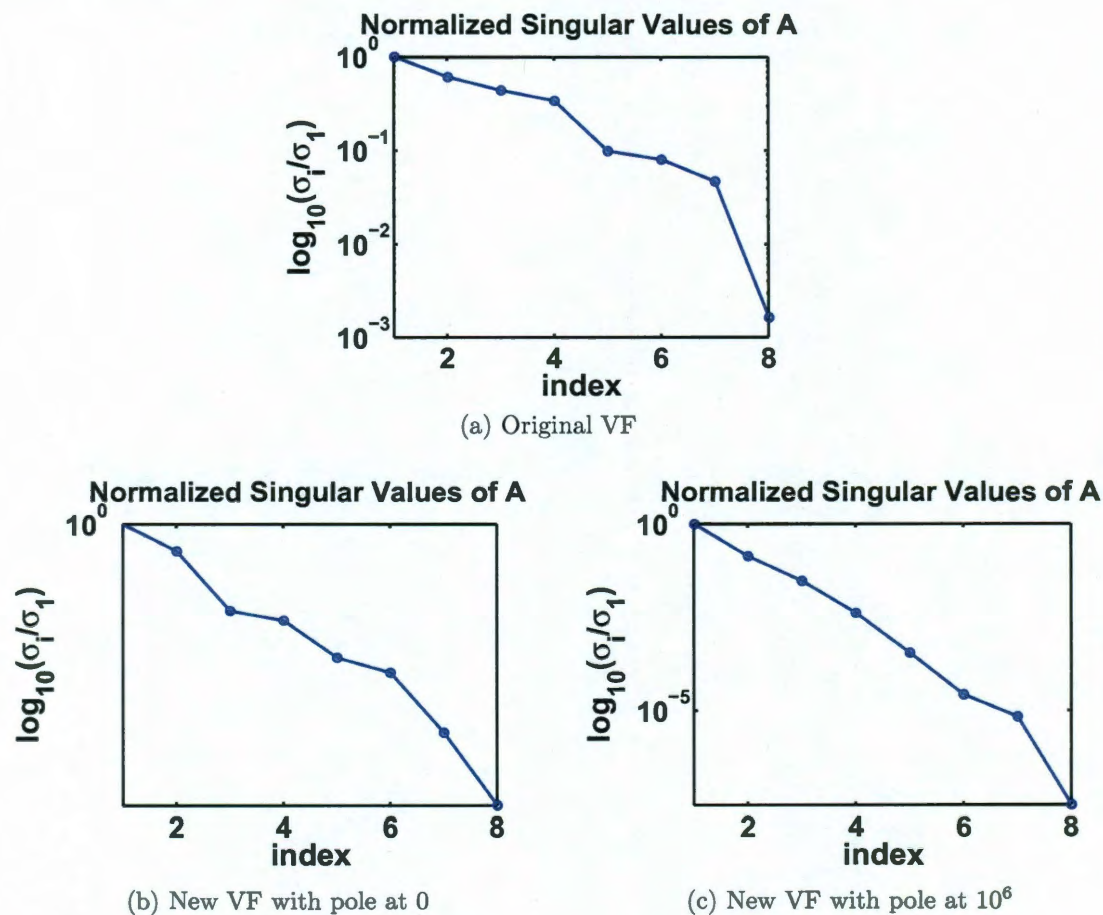


Figure 6.6: Normalized singular values of  $A$  when the model order is chosen 3 and noise is added to the measurements

Tables 6.5, 6.6, 6.7, 6.8 and 6.9 show the poles, residues and the  $D$ -term, respectively, which were recovered with each method. The poles and  $D$ -term closest to the original ones were obtained with VARPRO. The closest residue to the original residue located at 2 is obtained with the proposed reformulation with an extra pole at 0, while the closest residues to the original ones located at  $30 \pm 40j$  were obtained with VARPRO.

Original VF	New VF with extra pole at 0	New VF with extra pole at $10^6$	VARPRO
-4.9303	-4.7609	-4.9297	-4.9376

Table 6.5: Pole at  $-5$ 

Original VF	New VF with extra pole at 0	New VF with extra pole at $10^6$	VARPRO
$-98.647 \pm 497.96j$	$-98.627 \pm 491.95j$	$-98.645 \pm 497.94j$	$-98.843 \pm 498.06j$

Table 6.6: Poles at  $-100 \pm 500j$ 

Original VF	New VF with extra pole at 0	New VF with extra pole at $10^6$	VARPRO
2.0340	2.0229	2.0340	2.0351

Table 6.7: Residue at 2

Original VF	New VF with extra pole at 0	New VF with extra pole at $10^6$	VARPRO
$29.34 \pm 39.864j$	$27.759 \pm 40.049j$	$29.337 \pm 39.864j$	$29.404 \pm 39.904j$

Table 6.8: Residues at  $30 \pm 40j$ 

Original VF	New VF with extra pole at 0	New VF with extra pole at $10^6$	VARPRO
.50031	.50257	.50031	.50027

Table 6.9: D-term at .5

## Conclusion

---

This thesis started with an introduction in chapter 1, which discussed the problem of modeling systems from measurements of their frequency response from the point of view of electrical, as well as mechanical engineers. Next, chapter 2.1 presented a literature review summarizing the main directions considered to tackle this problem. Chapter 2.2 explained the three main contributions of this thesis:

1. addressing the issue of a large number of ports by employing tangential instead of matrix interpolation in the Loewner framework,
2. addressing the issue of parametric macromodeling by generalizing the Loewner matrix to two-variable transfer functions,
3. providing a convergence analysis of the already existing vector fitting algorithm, as well as several possible reformulations.

Before providing details about each contribution, chapter 3 consisted of a review of system theoretic definitions and basic concepts used in the next chapters.

The first main contribution is discussed in chapter 4. After a short review treating the derivation of the Loewner matrix for the scalar case based on Lagrange polynomials, followed by the introduction of the concept of tangential interpolation, and

the corresponding Loewner matrix for this case, the Loewner framework is completed with the introduction of the shifted Loewner matrix. As a minor contribution of this thesis, in Sect. 4.1.4, the concept of the Loewner matrix pencil constructed in the framework of tangential interpolation is analyzed from a theoretical point of view, by showing that a system can be recovered from tangential interpolation data given that the right amount of measurements are available and that the sampling directions are chosen appropriately. Sect. 4.2 shows various implementation approaches possible for the Loewner framework derived from the theoretical concepts described earlier. These proposed implementations are validated in Sect. 4.4 using numerical examples spanning various application areas. These approaches are especially suited for modeling devices with a large number of ports. We use no heuristics, but only the available data, and we are able to identify the order of the system. When the provided measurements are noisy and the noise level is high, we notice that the resulting models are poor, so Sect. 4.3.2 proposes an improvement. The main idea is to overmodel and apply a reduction step, in which the necessary information is extracted from the most dominant poles.

Chapter 5 discusses the second main contribution. Our approach to solving the parametric macromodeling problem is to generalize the Loewner matrix to the two variable case in Sect. 5.1 - 5.5. Using the frequency response measurements obtained for several parameter values, we generalize the *Loewner matrix* to the two variable case to construct models which are reduced both with respect to frequency and to the parameter. This generalization is valid for SISO, as well as MIMO systems and it was validated on academic, as well as practical examples consisting of real measurements.

Last, but not least, the third contribution of the thesis is presented in chapter 6: the first ever analysis of the convergence properties of the pole relocation iteration of the vector fitting algorithm. We found that for noise-free measurements and provided that the number of measurements is sufficient and the order of the model is chosen

equal to or higher than the order of the underlying system, vector fitting converges in 2 steps. For noisy measurements, the iteration may exhibit no convergence or more than one convergence points, situations which are clearly undesirable. To ensure that the iteration always converges, we propose to incorporate the Newton step in the vector fitting iteration. To save computational time by omitting the residue identification stage of vector fitting, we present a realization of the model that can be determined using the quantities found in the pole identification step (irrespective of whether convergence was achieved, or not). Based on insights from the Loewner framework, a reformulation presented in Sect. 6.7 proposes using an additional starting pole and, moreover, provides an educated guess for the starting poles at the first step of the pole relocation iteration.

By using polyharmonic distortion, nonlinear systems are characterized by X-parameters, which represent a generalization of S-parameters [80]. The X-parameters are measured by nonlinear vector network analyzers, and the problem amounts now to constructing a nonlinear model from the tabulated X-parameters. Future work includes extending the Loewner framework to modeling nonlinear systems described by X-parameters.

In terms of our solution to parametric macromodeling, future work is needed to generalize the shifted Loewner matrix to the case when the available data depends on more than one parameter. Moreover, this thesis only treated the two-variable generalization of the Loewner matrix, so our goal is to extend it to more than two variables (i.e., more than one design parameter).

## References

---

- [1] L. P. de Oliveira, K. Janssens, P. Gajdatsy, H. V. der Auweraer, P. S. Varoto, P. Sas, and W. Desmet, "Active sound quality control of engine induced cavity noise," *Mechanical Systems and Signal Processing*, vol. 23, no. 2, pp. 476 – 488, 2009. 8
- [2] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. Philadelphia: SIAM, 2005. 9, 18, 23, 27, 134
- [3] D. M. B. Van Overschee P., *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996. 12, 15
- [4] P. V. Overschee and B. D. Moor, "Continuous-time frequency domain subspace system identification," *Signal Processing*, vol. 52, no. 2, pp. 179–194, Jul. 1996. 12
- [5] C. Coelho, L. Silveira, and J. Phillips, "Passive constrained rational approximation algorithm using Nevanlinna-Pick interpolation," in *Proc. Design, Automation and Test Conf. in Europe*, 2002, pp. 923–930. 13, 46
- [6] A. C. Antoulas, "On the construction of passive models from frequency response data," *Automatisierungstechnik*, vol. 56, pp. 447–452, Aug. 2008. 13, 46
- [7] S.-H. Min and M. Swaminathan, "Construction of broadband passive macro-models from frequency data for simulation of distributed interconnect networks," *IEEE Trans. Electromagn. Compat.*, vol. 46, no. 4, pp. 544–558, Nov. 2004. 13
- [8] R. Gao, Y. Mekonnen, W. Beyene, and J. Schutt-Aine, "Black-box modeling of passive systems by rational function approximation," *IEEE Trans. Adv. Packag.*, vol. 28, no. 2, pp. 209–215, May 2005. 13
- [9] B. Peeters, H. Van der Auweraer, P. Guillaume, and J. Leuridan, "The PolyMAX frequency-domain method: a new standard for modal parameter estimation?" *Shock and Vibration*, vol. 11, no. 3/4, pp. 395 – 409, 2004. 13, 51



- [10] F. Lembregts, J. Leuridan, L. Zang, and H. Kanda, "Multiple input modal analysis of frequency response functions based on direct parameter identification," in *Proceedings of the International Modal Analysis Conference IMAC*, 1986, pp. 589–598. 13
- [11] D. L. Brown, R. J. Allemang, R. Zimmerman, and M. Mergeay, "Parameter estimation techniques for modal analysis," *Society of Automotive Engineers*, 1979, paper No. 790221. 13
- [12] R. Pintelon, P. Guillaume, Y. Rolain, J. Schoukens, and H. Van Hamme, "Parametric identification of transfer functions in the frequency domain—a survey," *IEEE Trans. Autom. Control*, vol. 39, no. 11, pp. 2245–2260, Nov. 1994. 13
- [13] D. Traina, G. Macchiarella, and T. K. Sarkar, "New general formulation of the cauchy method for the accurate model extraction of higher order microwave systems," *Microwave and Optical Technology Letters*, vol. 49, no. 8, pp. 1957–1961, 2007. 13
- [14] —, "Robust formulations of the cauchy method suitable for microwave duplexers modeling," *IEEE Trans. Microw. Theory Tech.*, vol. 55, no. 5, pp. 974–982, May 2007. 13, 14
- [15] A. G. Lamperez, T. K. Sarkar, and M. S. Palma, "Filter model generation from scattering parameters using the cauchy method," in *European Microwave Conf.*, sept. 2002, pp. 1–4. 13
- [16] A. Lamperez, T. Sarkar, and M. Palma, "Generation of accurate rational models of lossy systems using the cauchy method," *IEEE Microw. Wireless Compon. Lett.*, vol. 14, no. 10, pp. 490–492, oct. 2004. 13
- [17] R. Adve, T. Sarkar, S. Rao, E. Miller, and D. Pflug, "Application of the cauchy method for extrapolating/interpolating narrowband system responses," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 5, pp. 837–845, may 1997. 13, 14
- [18] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, pp. 1052–1061, Jul. 1999. 14, 15, 57, 104, 110, 112, 118, 119
- [19] —, "A robust approach for system identification in the frequency domain," *IEEE Trans. Power Del.*, vol. 19, pp. 1167–1173, Jul. 2004. 14, 17, 59, 104, 112
- [20] B. Gustavsen, "Improving the pole relocation properties of vector fitting," *IEEE Trans. Power Del.*, vol. 21, no. 3, pp. 1587–1592, Jul. 2006. 14, 57
- [21] S. Grivet-Talocia and M. Bandinu, "Improving the convergence of vector fitting for equivalent circuit extraction from noisy frequency responses," *IEEE Trans. Electromagn. Compat.*, vol. 48, pp. 104–120, Feb. 2006. 14, 15, 104, 110

- [22] D. Deschrijver, "Broadband macromodeling of linear systems by vector fitting," Ph.D. dissertation, Universiteit Antwerpen, Oct. 2007. 14, 104
- [23] F. Ebert and T. Stykel, "Rational interpolation, minimal realization and model reduction," DFG Research Center MATHEON, Tech. Rep. 371-2007, 2007. 14, 15, 60, 104, 109
- [24] G. Antonini, D. Deschrijver, and T. Dhaene, "A comparative study of vector fitting and orthonormal vector fitting techniques for EMC applications," *IEEE International Symposium on Electromagnetic Compatibility*, vol. 1, pp. 6–11, Aug. 2006. 14
- [25] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *IEEE Microw. Wireless Compon. Lett.*, vol. 18, no. 6, pp. 383–385, June 2008. 14, 57, 108
- [26] D. Deschrijver, B. Haegeman, and T. Dhaene, "Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses," *IEEE Trans. Adv. Packag.*, vol. 30, no. 2, pp. 216–225, May 2007. 14
- [27] S. Grivet-Talocia, "Package macromodeling via time-domain vector fitting," *IEEE Microw. Wireless Compon. Lett.*, vol. 13, no. 11, pp. 472–474, Nov. 2003. 14
- [28] W. Hendrickx and T. Dhaene, "A discussion of "Rational approximation of frequency domain responses by vector fitting",," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 441–443, Feb. 2006. 14, 112
- [29] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973. 15, 113, 114, 128
- [30] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," in *Institute of Physics, Inverse Problems*, 2002, pp. 1–26. 15, 113, 114
- [31] P. Triverio, S. Grivet-Talocia, and M. Nakhla, "A parameterized macromodeling strategy with uniform stability test," *IEEE Trans. Adv. Packag.*, vol. 32, no. 1, pp. 205–215, Feb. 2009. 15
- [32] —, "On the construction of uniformly stable multivariate interconnect macromodels," in *IEEE Workshop on Signal Propagation on Interconnects*, May 2009, pp. 1–4. 15
- [33] P. Triverio, M. Nakhla, and S. Grivet-Talocia, "Parametric macromodeling of multiport networks from tabulated data," in *Proc. IEEE Conf. on Electrical Performance of Electronic Packaging*, Oct 2007, pp. 51–54. 15

- [34] P. Triverio, S. Grivet-Talocia, and M. Nakhla, "An improved fitting algorithm for parametric macromodeling from tabulated data," in *IEEE Workshop on Signal Propagation on Interconnects*, May 2008, pp. 1–4. 15
- [35] D. Deschrijver, T. Dhaene, and D. De Zutter, "Robust parametric macromodeling using multivariate orthonormal vector fitting," *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 7, pp. 1661–1667, July 2008. 16
- [36] D. Deschrijver and T. Dhaene, "Stability and passivity enforcement of parametric macromodels in time and frequency domain," *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 11, pp. 2435–2441, Nov. 2008. 16
- [37] T. Dhaene and D. Deschrijver, "Stable parametric macromodeling using a recursive implementation of the vector fitting algorithm," *IEEE Microw. Wireless Compon. Lett.*, vol. 19, no. 2, pp. 59–61, Feb. 2009. 16
- [38] F. Ferranti, D. Deschrijver, L. Knockaert, and T. Dhaene, "Hybrid algorithm for compact and stable macromodelling of parameterised frequency responses," *Electronics Letters*, vol. 45, no. 10, pp. 493–495, July 2009. 16
- [39] ———, "Fast parametric macromodeling of frequency responses using parameter derivatives," *IEEE Microw. Wireless Compon. Lett.*, no. 12, pp. 761–763, 2008. 16
- [40] B. Bond and L. Daniel, "Parameterized model order reduction of nonlinear dynamical systems," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2005, pp. 487 – 494. 16
- [41] L. Daniel, O. C. Siong, L. Chay, K. H. Lee, and J. White, "A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 5, pp. 678 – 693, May 2004. 16
- [42] U. Baur, C. Beattie, P. Benner, and S. Gugercin, "Interpolatory projection methods for parameterized model reduction," TU Chemnitz, Tech. Rep., Nov 2009. 16
- [43] S. Lefteriu and A. C. Antoulas, "A new approach to modeling multi-port systems from frequency domain data," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 1, pp. 14 –27, Jan. 2010. 16, 17, 25, 31, 40, 41, 42, 45, 46, 57, 59, 93
- [44] A. J. Mayo and A. C. Antoulas, "A framework for the solution of the generalized realization problem," *Linear Algebra and Its Applications*, vol. 405, no. 2-3, pp. 634–662, 2007. 18, 25, 27, 29, 30, 32, 44, 93, 134
- [45] E. Yip and R. Sincovec, "Solvability, controllability, and observability of continuous descriptor systems," *IEEE Trans. Autom. Control*, vol. 26, no. 3, pp. 702–707, Jun. 1981. 22

- [46] S. Lefteriu, "New approaches to modeling multi-port scattering parameters," Master's thesis, Rice University, Houston, TX, 2008. 25, 45, 109, 110
- [47] S. Lefteriu and A. C. Antoulas, "A new adaptive approach to modeling measured multi-port scattering parameters," in *Scientific Computing in Electrical Engineering SCEE 2008*, ser. Mathematics in Industry, J. Roos and L. R. Costa, Eds. Springer, 2010, vol. 14, pp. 21–28. 25, 45, 57
- [48] S. Lefteriu and A. Antoulas, "Modeling multi-port systems from frequency response data via tangential interpolation," in *IEEE Workshop on Signal Propagation on Interconnects*, May 2009, pp. 1–4. 25
- [49] S. Lefteriu, A. Ionita, and A. Antoulas, "Modeling systems based on noisy frequency and time domain measurements," in *Perspectives in Mathematical System Theory, Control, and Signal Processing*, ser. Lecture Notes in Control and Information Sciences, J. Willems, S. Hara, Y. Ohta, and H. Fujioka, Eds. Springer, 2010, vol. 398, pp. 365–378. 25, 48, 49
- [50] S. Lefteriu and A. Antoulas, "Topics in model order reduction with applications to circuit simulation," in *Model Reduction for Circuit Simulation*, ser. Lecture Notes in Electrical Engineering, H. M. Benner, P. and E. J. W. ter Maten, Eds., vol. 13. Springer, 2011. 25, 32, 33, 36, 39, 40, 168, 172, 175
- [51] A. Antoulas, A. Ionita, and S. Lefteriu, "On two-variable rational interpolation," *Linear Algebra and its Applications*, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0024379511005295> 26, 74, 77, 132
- [52] A. C. Antoulas and B. D. O. Anderson, "On the scalar rational interpolation problem," *IMA J. of Mathematical Control and Information*, vol. 3, no. 2-3, pp. 61–88, 1986, special issue on Parametrization problems. 27, 28, 134
- [53] B. D. O. Anderson and A. C. Antoulas, "Rational interpolation and state-variable realizations," *Linear Algebra and Its Applications*, vol. 137/138, pp. 479–509, 1990, special issue on Matrix problems. 27, 28, 31, 134
- [54] M. Kuijper and J. M. Schumacher, "Realization of autoregressive equations in pencil and descriptor form," *SIAM J. Control Optim.*, vol. 28, no. 5, pp. 1162–1189, 1990. 28
- [55] A. Antoulas and B. Anderson, "State space and polynomial approaches to rational interpolation," in *Realization and Modeling in System Theory*, M. Kaashoek, J. van Schuppen, and A. Ran, Eds. Birkhäuser, 1990, pp. 73–82. 28
- [56] K. Gallivan, A. Vandendorpe, and P. V. Dooren, "Model reduction of MIMO systems via tangential interpolation," *SIAM J. Matrix Anal. Appl.*, vol. 26, pp. 328–349, Feb. 2005. 29

- [57] T. Kailath, *Linear Systems*. Prentice-Hall, 1979. 36
- [58] G. W. Stewart, "Perturbation theory for the singular value decomposition," in *SVD and Signal Processing, II: Algorithms, Analysis and Applications*. Elsevier, 1990, pp. 99–109. 45
- [59] J. Demmel and B. Kågström, "The generalized Schur decomposition of an arbitrary pencil  $A - zB$ : Robust software with error bounds and applications. Part I: Theory and algorithms," *ACM Trans. Math. Softw.*, vol. 19, no. 2, pp. 160–174, 1993. 45
- [60] —, "The generalized Schur decomposition of an arbitrary pencil  $A - zB$ : Robust software with error bounds and applications. Part II: Software and applications," *ACM Trans. Math. Softw.*, vol. 19, no. 2, pp. 175–201, 1993. 45
- [61] S. Grivet-Talocia, "Passivity enforcement via perturbation of Hamiltonian matrices," *IEEE Trans. Circuits Syst. I*, vol. 51, pp. 1755–1769, Sep. 2004. 46
- [62] H. Van der Auweraer and B. Peeters, "Discriminating physical poles from mathematical poles in high order systems: use and automation of the stabilization diagram," in *Proc. IEEE Instrumentation and Measurement Technology Conference*, vol. 3, May 2004, pp. 2193 – 2198 Vol.3. 51
- [63] T. G. Wright, "EigTool," 2002, software available at <http://www.comlab.ox.ac.uk/projects/pseudospectra/eigtool>. 55
- [64] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: the Behaviour of Nonnormal Matrices and Operators*. Philadelphia: SIAM, 2005. 55
- [65] R. Ionutiu, J. Rommes, and W. H. A. Schilders, "SparseRC: sparsity preserving model reduction for RC circuits with many terminals," Technische Universiteit Eindhoven, CASA Report 11-05, 2011, submitted to IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 63
- [66] F. Yang, X. Zeng, Y. Su, and D. Zhou, "RLC equivalent circuit synthesis method for structure-preserved reduced-order model of interconnect in VLSI," *Communications in computational physics*, vol. 3, no. 2, pp. 376–396, 2008. 64
- [67] R. Ionutiu and J. Rommes, "On synthesis of reduced order models," in *Model Reduction for Circuit Simulation*, ser. Lecture Notes in Electrical Engineering, P. Benner, M. Hinze, and J. ter Maten, Eds. Berlin/Heidelberg: Springer-Verlag, 2010. 64
- [68] Cadence, "Spectre," <http://www.cadence.com>. 64
- [69] R. Ionutiu, "Model reduction for multi-terminal systems with applications to circuit simulation," Ph.D. dissertation, Jacobs University Bremen and Technische Universiteit Eindhoven, 2011, in preparation. 66

- [70] H. Antil, M. Heinkenschloss, and R. Hoppe, "Domain decomposition and balanced truncation model reduction for shape optimization of the Stokes system," *Optim. Meth. & Software*, pp. 81–100, 2010. 66, 67
- [71] H. Antil, M. Heinkenschloss, R. H. Hoppe, C. Linsenmann, and A. Wixforth, "Reduced order modeling based shape optimization of surface acoustic wave driven microfluidic biochips," *Mathematics and Computers in Simulation*, 2010, in Press. 66, 67
- [72] D. Köster, "Numerical simulation of acoustic streaming on surface acoustic wave-driven biochips," *SIAM J. Sci. Comput.*, vol. 29, pp. 2352–2380, Oct. 2007. 66, 67
- [73] S. Lefteriu, A. Antoulas, and A. Ionita, "Parametric model order reduction from measurements," in *Proc. IEEE Conf. Electrical Performance of Electronic Packaging and Systems*, Oct. 2010, pp. 193–196. 74, 77, 93
- [74] V. Mehrmann and T. Stykel, "Balanced truncation model reduction for large-scale systems in descriptor form," in *Dimension Reduction of Large-Scale Systems*, ser. Lecture Notes in Computational Science and Engineering, T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, T. Schlick, P. Benner, D. C. Sorensen, and V. Mehrmann, Eds. Springer, 2005, pp. 83–115. 78
- [75] S. Lefteriu, A. Antoulas, and A. Ionita, "Parametric model reduction in the Loewner framework," in *International Federation of Automatic Control*, 2011, accepted. 84, 94
- [76] P. Triverio, M. Nakhla, and S. Grivet-Talocia, "Parametric macromodeling of multiport networks from tabulated data," in *Proc. IEEE Conf. Electrical Performance of Electronic Packaging*, Oct. 2007, pp. 51–54. 92
- [77] T. Breiten and P. Benner, "Nonlinear model reduction via quadratic bilinear control systems." [Online]. Available: [www3.math.tu-berlin.de/modred2010/Abstracts/Breiten\\_Tobias\\_abstract.pdf](http://www3.math.tu-berlin.de/modred2010/Abstracts/Breiten_Tobias_abstract.pdf) 103
- [78] C. Lei, Y. Wang, Q. Chen, and N. Wong, "A decade of vector fitting development: Applications on signal/power integrity," *AIP Conference Proceedings*, vol. 1285, no. 1, pp. 435–449, 2010. [Online]. Available: <http://link.aip.org/link/?APC/1285/435/1> 112
- [79] L. Kaufman and G. Sylvester, "Separable nonlinear least squares with multiple right-hand sides," *SIAM J. Matrix Anal. Appl.*, vol. 13, pp. 68–89, January 1992. 115
- [80] M. Heimlich, "Get on the same nonlinear page," *IEEE Microwave Magazine*, vol. 12, no. 2, pp. 32–37, 2011. 145

- [81] T. Reis, "Systems theoretic aspects of PDAEs and applications to electrical circuits," Ph.D. dissertation, Technische Universität Kaiserslautern, 2006. 168
- [82] A. Schneider, *Matrix decomposition based approaches for model order reduction of linear systems with a large number of terminals*. 172
- [83] P. Feldmann, "Model order reduction techniques for linear systems with large numbers of terminals," *Proc. Design, Automation and Test Conf. in Europe*, pp. 944–947, 2004. 172
- [84] P. Feldmann and F. Liu, "Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2004, pp. 88–92. 172
- [85] Y. Wang, C. U. Lei, G. Pang, and N. Wong, "MFTI: Matrix-format tangential interpolation for modeling multi-port systems," in *Proc. ACM/IEEE Design Automation Conference*, 2010, pp. 683–686. 172
- [86] J. Hokanson, "Reduced variable projection for fast complex exponential fitting," in preparation. 186
- [87] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*. SIAM, 2003. 191

### A.1 Proof of Theorem 5.4.2

*Proof.* We need to show that  $\mathbf{H}(\lambda_i, \pi_j) = \mathbf{W}_{i,j}$  and  $\mathbf{H}(\mu_k, \nu_l) = \mathbf{V}_{kl}$ . We begin with the condition  $\mathbf{H}(\lambda_i, \pi_j) = \mathbf{C}\Phi^{-1}(\lambda_i, \pi_j)\mathbf{B} = \mathbf{W}_{i,j}$ ,  $\forall i = 0, \dots, k_1, \forall j = 0, \dots, k_2$ . Let us denote  $\Phi^{-1}(\lambda_i, \pi_j)\mathbf{B}$  as  $\mathbf{v}$ , so  $\Phi(\lambda_i, \pi_j)\mathbf{v} = \mathbf{B}$ :

$$\begin{array}{|c|c|c|c|}
 \hline
 (\lambda_i - \lambda_0)\mathbf{I}_p & (\lambda_1 - \lambda_i)\mathbf{I}_p & & \\
 \vdots & \ddots & & \\
 (\lambda_i - \lambda_0)\mathbf{I}_p & & (\lambda_{k_1} - \lambda_i)\mathbf{I}_p & \\
 \hline
 \alpha_{00} & \dots & \alpha_{k_1 0} & (\pi_j - \pi_0)\mathbf{I}_p \dots (\pi_j - \pi_0)\mathbf{I}_p \\
 \alpha_{01} & \dots & \alpha_{k_1 1} & (\pi_1 - \pi_j)\mathbf{I}_p \\
 \vdots & \ddots & \vdots & \ddots \quad \vdots \\
 \alpha_{0k_2} & \dots & \alpha_{k_1 k_2} & (\pi_{k_2} - \pi_j)\mathbf{I}_p \\
 \hline
 \beta_{00} & \dots & \beta_{k_1 0} & (\pi_j - \pi_0)\mathbf{I}_p \dots (\pi_j - \pi_0)\mathbf{I}_p \quad q_0\mathbf{I}_p \\
 \beta_{01} & \dots & \beta_{k_1 1} & (\pi_1 - \pi_j)\mathbf{I}_p \quad q_1\mathbf{I}_p \\
 \vdots & \ddots & \vdots & \ddots \quad \vdots \\
 \beta_{0,k_2} & \dots & \beta_{k_1,k_2} & (\pi_{k_2} - \pi_j)\mathbf{I}_p \quad q_{k_2}\mathbf{I}_p \\
 \hline
 \end{array}
 \begin{array}{c}
 \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_{k_1} \end{bmatrix} \\
 \mathbf{y}_1 \\
 \vdots \\
 \mathbf{y}_{k_2} \\
 \mathbf{z}_0 \\
 \vdots \\
 \mathbf{z}_{k_2}
 \end{array}
 =
 \begin{array}{c}
 \begin{bmatrix} \mathbf{0}_p \\ \vdots \\ \mathbf{0}_p \\ q_0\mathbf{I}_p \\ \vdots \\ q_{k_2}\mathbf{I}_p \\ \mathbf{0}_p \\ \vdots \\ \mathbf{0}_p \end{bmatrix}
 \end{array}
 \tag{A.1}$$



Therefore,

$$\Rightarrow \left\{ \begin{array}{l} (\lambda_i - \lambda_0)\mathbf{x}_0 = (\lambda_i - \lambda_1)\mathbf{x}_1 \Rightarrow \mathbf{x}_1 = \frac{\lambda_i - \lambda_0}{\lambda_i - \lambda_1}\mathbf{x}_0, \text{ unless } i=1 \\ (\lambda_i - \lambda_0)\mathbf{x}_0 = (\lambda_i - \lambda_2)\mathbf{x}_2 \Rightarrow \mathbf{x}_2 = \frac{\lambda_i - \lambda_0}{\lambda_i - \lambda_2}\mathbf{x}_0, \text{ unless } i=2 \\ \vdots \\ (\lambda_i - \lambda_0)\mathbf{x}_0 = (\lambda_i - \lambda_i)\mathbf{x}_i \Rightarrow \mathbf{x}_0 = \mathbf{0}, \\ \vdots \\ (\lambda_i - \lambda_0)\mathbf{x}_0 = (\lambda_i - \lambda_{k_1})\mathbf{x}_{k_1} \Rightarrow \mathbf{x}_{k_1} = \frac{\lambda_i - \lambda_0}{\lambda_i - \lambda_{k_1}}\mathbf{x}_0, \text{ unless } i = k_1 \\ \alpha_{00}\mathbf{x}_0 + \alpha_{10}\mathbf{x}_1 + \dots + \alpha_{k_1 0}\mathbf{x}_{k_1} + (\pi_j - \pi_0)(\mathbf{y}_1 + \dots + \mathbf{y}_{k_2}) = q_0\mathbf{I}_p \\ \alpha_{01}\mathbf{x}_0 + \alpha_{11}\mathbf{x}_1 + \dots + \alpha_{k_1 1}\mathbf{x}_{k_1} + (\pi_1 - \pi_j)\mathbf{y}_1 = q_0\mathbf{I}_p \\ \vdots \\ \alpha_{0j}\mathbf{x}_0 + \alpha_{1j}\mathbf{x}_1 + \dots + \alpha_{k_1 j}\mathbf{x}_{k_1} + (\pi_j - \pi_j)\mathbf{y}_j = q_j\mathbf{I}_p \\ \vdots \\ \alpha_{0k_2}\mathbf{x}_0 + \alpha_{1k_2}\mathbf{x}_1 + \dots + \alpha_{k_1 k_2}\mathbf{x}_{k_1} + (\pi_{k_2} - \pi_j)\mathbf{y}_1 = q_{k_2}\mathbf{I}_p \\ \beta_{00}\mathbf{x}_0 + \beta_{10}\mathbf{x}_1 + \dots + \beta_{k_1 0}\mathbf{x}_{k_1} + (\pi_j - \pi_0)(\mathbf{z}_0 + \dots + \mathbf{z}_{k_2-1}) + q_0\mathbf{z}_{k_2} = \mathbf{0} \\ \beta_{01}\mathbf{x}_0 + \beta_{11}\mathbf{x}_1 + \dots + \beta_{k_1 1}\mathbf{x}_{k_1} + (\pi_1 - \pi_j)\mathbf{z}_0 + q_1\mathbf{z}_{k_2} = \mathbf{0} \\ \vdots \\ \beta_{0j}\mathbf{x}_0 + \beta_{1j}\mathbf{x}_1 + \dots + \beta_{k_1 j}\mathbf{x}_{k_1} + (\pi_j - \pi_j)\mathbf{z}_0 + q_j\mathbf{z}_{k_2} = \mathbf{0} \\ \vdots \\ \beta_{0k_2}\mathbf{x}_0 + \beta_{1k_2}\mathbf{x}_1 + \dots + \beta_{k_1 k_2}\mathbf{x}_{k_1} + (\pi_{k_2} - \pi_j)\mathbf{z}_0 + q_{k_2}\mathbf{z}_{k_2} = \mathbf{0}. \end{array} \right. \quad (\text{A.2})$$

Thus  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{k_1} = \mathbf{0}$  and  $\mathbf{x}_i \neq \mathbf{0}$ . This implies

$$\left\{ \begin{array}{l}
 \alpha_{i0}\mathbf{x}_i + (\pi_j - \pi_0)(\mathbf{y}_1 + \dots + \mathbf{y}_{k_2}) = q_0\mathbf{I}_p \\
 \alpha_{i1}\mathbf{x}_i + (\pi_1 - \pi_j)\mathbf{y}_1 = q_0\mathbf{I}_p \\
 \vdots \\
 \alpha_{ij}\mathbf{x}_i + (\pi_j - \pi_j)\mathbf{y}_j = q_j\mathbf{I}_p \Rightarrow \mathbf{x}_i = \alpha_{ij}^{-1}q_j\mathbf{I}_p \\
 \vdots \\
 \alpha_{ik_2}\mathbf{x}_i + (\pi_{k_2} - \pi_j)\mathbf{y}_1 = q_{k_2}\mathbf{I}_p \\
 \beta_{i0}\mathbf{x}_i + (\pi_j - \pi_0)(\mathbf{z}_0 + \dots + \mathbf{z}_{k_2-1}) + q_0\mathbf{z}_{k_2} = \mathbf{0} \\
 \beta_{i1}\mathbf{x}_i + (\pi_1 - \pi_j)\mathbf{z}_0 + q_1\mathbf{z}_{k_2} = \mathbf{0} \\
 \vdots \\
 \beta_{ij}\mathbf{x}_i + q_j\mathbf{z}_{k_2} = \mathbf{0} \Rightarrow (\mathbf{W}_{ij}\alpha_{ij}\alpha_{ij}^{-1} + \mathbf{z}_{k_2})q_j = \mathbf{0} \Rightarrow -\mathbf{z}_{k_2} = \mathbf{C}\Phi^{-1}(\lambda_i, \pi_j)\mathbf{B} = \mathbf{W}_{ij} \\
 \vdots \\
 \beta_{ik_2}\mathbf{x}_i + (\pi_{k_2} - \pi_j)\mathbf{z}_0 + q_{k_2}\mathbf{z}_{k_2} = \mathbf{0}.
 \end{array} \right. \quad (\text{A.3})$$

Note that this part of the proof holds for any  $\alpha_{ij}$  invertible as we made use of no other properties of the matrices  $\alpha_{ij}$ . To show that the second condition also holds, namely  $\mathbf{H}(\mu_k, \nu_l) = \mathbf{C}\hat{\Phi}^{-1}(\mu_k, \nu_l)\mathbf{B} = \mathbf{V}_{kl}$ ,  $\forall k = 0, \dots, k_1$ ,  $\forall l = 0, \dots, k_2$ , let us denote  $\Phi^{-1}(\mu_k, \nu_l)\mathbf{B}$  as  $\mathbf{v}$ , so  $\Phi(\mu_k, \nu_l)\mathbf{v} = \mathbf{B}$ :

$$\begin{array}{|c|c|c|c|}
 \hline
 (\mu_k - \lambda_0)\mathbf{I}_p & (\lambda_1 - \mu_k)\mathbf{I}_p & & \\
 \vdots & \ddots & & \\
 (\mu_k - \lambda_0)\mathbf{I}_p & (\lambda_{k_1} - \mu_k)\mathbf{I}_p & & \\
 \hline
 \alpha_{00} & \dots & \alpha_{k_1 0} & (\nu_l - \pi_0)\mathbf{I}_p \dots (\nu_l - \pi_0)\mathbf{I}_p \\
 \alpha_{01} & \dots & \alpha_{k_1 1} & (\pi_1 - \nu_l)\mathbf{I}_p \\
 \vdots & \ddots & \vdots & \ddots \quad \vdots \\
 \alpha_{0k_2} & \dots & \alpha_{k_1 k_2} & (\pi_{k_2} - \nu_l)\mathbf{I}_p \\
 \hline
 \beta_{00} & \dots & \beta_{k_1 0} & (\nu_l - \pi_0)\mathbf{I}_p \dots (\nu_l - \pi_0)\mathbf{I}_p \quad q_0\mathbf{I}_p \\
 \beta_{01} & \dots & \beta_{k_1 1} & (\pi_1 - \nu_l)\mathbf{I}_p \quad q_1\mathbf{I}_p \\
 \vdots & \ddots & \vdots & \ddots \quad \vdots \quad \vdots \\
 \beta_{0, k_2} & \dots & \beta_{k_1, k_2} & (\pi_{k_2} - \nu_l)\mathbf{I}_p \quad q_{k_2}\mathbf{I}_p \\
 \hline
 \end{array} = \begin{array}{|c|}
 \hline
 \mathbf{x}_0 \\
 \vdots \\
 \mathbf{x}_{k_1} \\
 \hline
 \mathbf{y}_1 \\
 \vdots \\
 \mathbf{y}_{k_2} \\
 \hline
 \mathbf{z}_0 \\
 \vdots \\
 \mathbf{z}_{k_2} \\
 \hline
 \end{array} = \begin{array}{|c|}
 \hline
 \mathbf{0}_p \\
 \vdots \\
 \mathbf{0}_p \\
 \hline
 q_0\mathbf{I}_p \\
 \vdots \\
 q_{k_2}\mathbf{I}_p \\
 \hline
 \mathbf{0}_p \\
 \vdots \\
 \mathbf{0}_p \\
 \hline
 \end{array} \quad (\text{A.4})$$

Recall that the vector  $\left[ \alpha_{00} \dots \alpha_{0k_2} \dots \alpha_{k_1 0} \dots \alpha_{k_1 k_2} \right]^T$  is in the nullspace of the matrix  $\mathbb{L}^{(M)}$  so multiplying each row of  $\mathbb{L}^{(M)}$  by this vector yields

$$\sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\mathbf{V}_{kl} - \mathbf{W}_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)} \alpha_{ij} = \mathbf{0}, \quad \forall k = 0, \dots, k_1, \quad \forall l = 0, \dots, k_2.$$

Therefore,

$$\mathbf{V}_{kl} \sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\alpha_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)} = \sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\mathbf{W}_{ij} \alpha_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)} = \sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\beta_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)}. \quad (\text{A.5})$$

Summing up the expressions in  $\alpha$  and  $\beta$ , respectively, from Eq.(A.4), we obtain

$$\sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\alpha_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)} \mathbf{x}_0 + \underbrace{\frac{1}{\mu_k - \lambda_0} \sum_{j=1}^{k_2} \mathbf{y}_j - \frac{1}{\mu_k - \lambda_0} \sum_{j=1}^{k_2} \mathbf{y}_j}_{\mathbf{0}} = \frac{1}{\mu_k - \lambda_0} \sum_{j=0}^{k_2} \frac{q_{j+1}}{\nu_l - \pi_j} \mathbf{I}_p \quad (\text{A.6})$$

$$\sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\beta_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)} \mathbf{x}_0 + \underbrace{\frac{1}{\mu_k - \lambda_0} \sum_{j=0}^{k_2-1} \mathbf{z}_j - \frac{1}{\mu_k - \lambda_0} \sum_{j=0}^{k_2-1} \mathbf{z}_j + \frac{\mathbf{z}_{k_2}}{\mu_k - \lambda_0} \sum_{j=0}^{k_2} \frac{q_{j+1}}{\nu_l - \pi_j}}_{\mathbf{0}} \mathbf{I}_p = \mathbf{0} \quad (\text{A.7})$$

Substituting Eq.(A.5) and Eq.(A.6) into Eq.(A.7), we obtain

$$\mathbf{V}_{kl} \sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\alpha_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)} \mathbf{x}_0 + \mathbf{z}_{k_2} \sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\alpha_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)} \mathbf{x}_0 = \mathbf{0} \quad (\text{A.8})$$

$$\Rightarrow (\mathbf{V}_{kl} + \mathbf{z}_{k_2}) \sum_{i=0}^{k_1} \sum_{j=0}^{k_2} \frac{\alpha_{ij}}{(\mu_k - \lambda_i)(\nu_l - \pi_j)} \mathbf{x}_0 = \mathbf{0}. \quad (\text{A.9})$$

Thus,  $\mathbf{V}_{kl} = -\mathbf{z}_{k_2} = \mathbf{C} \hat{\Phi}^{-1}(\mu_k, \nu_l) \mathbf{B}$ . □

## Pseudocodes

## B.1 SVD implementation in the Loewner framework

---

**Algorithm 1**  $(\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}) = \text{SVDapproach}(\mathbb{L}, \sigma\mathbb{L}, \mathbf{V}, \mathbf{W}, \omega)$

---

**Given:**  $\mathbb{L}, \sigma\mathbb{L}, \mathbf{V}, \mathbf{W}$  constructed as in Sect.4.2.1, 4.2.2 or 4.2.3 and  $\omega$ , the vector of measured frequencies scaled by  $2\pi$ .

**Output:**  $\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}$  such that  $\mathbf{H}(j\omega_i) \approx \mathbf{H}^{(i)}$ ,  $i = 1, \dots, k$ .

- 1:  $x \leftarrow j\omega_1$  (if Sect.4.2.1 was followed) or  $x \leftarrow \omega_1$  (if Sect.4.2.2 or 4.2.3 was followed).
  - 2:  $[\mathbf{Y}, \Sigma, \mathbf{X}] = \text{svd}(x\mathbb{L} - \sigma\mathbb{L})$ .
  - 3: Plot  $\log_{10}(\frac{\sigma_i}{\sigma_1})$ , where  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_k)$ ,  $\sigma_1 \geq \dots \geq \sigma_k$ .
  - 4: Choose  $n$  such that  $\log_{10}(\frac{\sigma_{n+1}}{\sigma_n}) \gg \log_{10}(\frac{\sigma_{n+2}}{\sigma_{n+1}})$ ,  $\log_{10}(\frac{\sigma_{n+1}}{\sigma_n}) \gg \log_{10}(\frac{\sigma_n}{\sigma_{n-1}})$ ,  $\sigma_{n+1} \approx \text{Noise} \cdot \sqrt{N}$ .
  - 5: Set  $\mathbf{Y}_1 \leftarrow \mathbf{Y}(:, 1:n)$ ,  $\mathbf{X}_1 \leftarrow \mathbf{X}(:, 1:n)$ .
  - 6:  $\mathbf{E} \leftarrow -\mathbf{Y}_1^* \mathbb{L} \mathbf{X}_1$ ,  $\mathbf{A} \leftarrow -\mathbf{Y}_1^* \sigma \mathbb{L} \mathbf{X}_1$ ,  $\mathbf{B} \leftarrow \mathbf{Y}_1^* \mathbf{V}$ ,  $\mathbf{C} \leftarrow \mathbf{W} \mathbf{X}_1$ .
- 

## B.2 Adaptive implementation in the Loewner framework

---

**Algorithm 2** ( $\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}$ ) = ADAPTIVEapproach( $\mathbf{H}, \omega$ )

---

**Given:**  $\mathbf{H}, \omega$  where  $\mathbf{H}^{(i)}$  is the measured frequency response at  $f_i = \frac{\omega_i}{2\pi}$ .**Output:**  $\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}$  such that  $\mathbf{H}(j\omega_i) \approx \mathbf{H}^{(i)}, i = 1, \dots, N$ .

- 1: Generate  $k$ , a set of  $p$  linearly distributed numbers between 1 and  $N$ , set  $m \leftarrow p$ .
  - 2: Construct  $\Lambda, M, \mathbf{L}, \mathbf{R}, \mathbf{V}, \mathbf{W}, \mathbb{L}, \sigma\mathbb{L}$  as described in Sect.4.2.1, using  $\omega_{k(1)}, \dots, \omega_{k(p)}$  and  $\mathbf{H}^{(k(1))}, \dots, \mathbf{H}^{(k(p))}$ , set  $\mathbf{E} \leftarrow -\mathbb{L}, \mathbf{A} \leftarrow -\sigma\mathbb{L}, \mathbf{B} \leftarrow \mathbf{V}, \mathbf{C} \leftarrow \mathbf{W}$ .
  - 3: **for**  $i = 1, \dots, N$  **do**
  - 4:    $\mathbf{H}_i \leftarrow \mathbf{C}(j \cdot \omega_i \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}$ .
  - 5:    $[\mathbf{Y}, \Sigma, \mathbf{X}] = \text{svd}(\mathbf{H}_i - \mathbf{H}^{(i)})$ ,  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_p)$ ,  $err_i \leftarrow \sigma_1$ .
  - 6: **end for**
  - 7: Sort  $err$  descendingly:  $[vmax, imax] = \text{sort}(err)$ , where  $vmax$  are the sorted values and  $imax$  are the sorted indices.
  - 8: **while**  $vmax(1) > th$ , where  $th = Noise \cdot \sqrt{N}$  **do**
  - 9:   Update  $\Lambda, M$  using  $\omega_{imax(1)}, \dots, \omega_{imax(p)}$ .
  - 10:   **for**  $i = 1, \dots, p$  **do**
  - 11:      $[\mathbf{Y}, \Sigma, \mathbf{X}] = \text{svd}(\mathbf{H}_{imax(i)} - \mathbf{H}^{(imax(i))})$ ,  $\mathbf{r}_{i+m} = \mathbf{X}(:, i)$ ,  $\mathbf{w}_{i+m} = \mathbf{H}^{(imax(i))} \mathbf{r}_{i+m}$ ,  $\ell_{i+m} = \mathbf{Y}(:, i)^T$ ,  $\mathbf{v}_{i+p} = \ell_{i+p} \mathbf{H}^{(imax(i))}$ .
  - 12:   **end for**
  - 13:   Update  $\mathbb{L}, \sigma\mathbb{L}$  using Eq.(4.21) and (4.23), set  $\mathbf{E} \leftarrow -\mathbb{L}, \mathbf{A} \leftarrow -\sigma\mathbb{L}, \mathbf{B} \leftarrow \mathbf{V}, \mathbf{C} \leftarrow \mathbf{W}$ , as well as  $m \leftarrow m + p$ .
  - 14:   Repeat Steps 3-7.
  - 15: **end while**
  - 16: Repeat Steps 9-13.
  - 17:  $x \leftarrow j\omega_1$  and  $[\mathbf{Y}, \Sigma, \mathbf{X}] = \text{svd}(x\mathbb{L} - \sigma\mathbb{L})$ .
  - 18: Plot  $\log_{10}(\frac{\sigma_i}{\sigma_1})$ , where  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_m)$ .
  - 19: Choose  $n$  such that  $\log_{10}(\frac{\sigma_{n+1}}{\sigma_n}) \gg \log_{10}(\frac{\sigma_{n+2}}{\sigma_{n+1}})$ ,  $\log_{10}(\frac{\sigma_{n+1}}{\sigma_n}) \gg \log_{10}(\frac{\sigma_n}{\sigma_{n-1}})$ ,  $\sigma_{n+1} \approx Noise \cdot \sqrt{N}$ .
  - 20: Set  $\mathbf{Y}_1 \leftarrow \mathbf{Y}(:, 1:n)$ ,  $\mathbf{X}_1 \leftarrow \mathbf{X}(:, 1:n)$ .
  - 21:  $\mathbf{E} \leftarrow -\mathbf{Y}_1^* \mathbb{L} \mathbf{X}_1$ ,  $\mathbf{A} \leftarrow -\mathbf{Y}_1^* \sigma \mathbb{L} \mathbf{X}_1$ ,  $\mathbf{B} \leftarrow \mathbf{Y}_1^* \mathbf{V}$ ,  $\mathbf{C} \leftarrow \mathbf{W} \mathbf{X}_1$ .
-

### B.3 Recursive Implementation of the Newton step in the VF iteration

---

**Algorithm 3**  $\mathbf{a} = \text{NEWTONonVF\_Recursive}(\mathbf{H}, \omega, \mathbf{a})$

---

**Given:**  $\mathbf{H}, \omega, \mathbf{a}$  where  $H_i$  is the measured frequency response at  $f_i = \frac{\omega_i}{2\pi}$ , and  $\mathbf{a}$ , the initial set of starting poles.

**Output:**  $\mathbf{a}$ , the set of starting poles at the final step.

- 1: Solve  $\mathbf{A}(s_i, N; a_i, n)\mathbf{x} \approx \mathbf{b}$  for  $\mathbf{x}$ , so  $\tilde{\mathbf{c}} = \mathbf{x}((n+2) : (2n+1))$ .
  - 2: **for**  $i = 1, \dots, n$  **do**
  - 3:     Compute  $\frac{\partial \mathbf{A}}{\partial a_i}$  using Eq.(6.45).
  - 4:     Solve Eq.(6.46) for  $\frac{\partial \mathbf{x}}{\partial a_i}$ , so  $\mathbf{J}_{\tilde{\mathbf{c}}}(:, i) = \frac{\partial \mathbf{x}}{\partial a_i}((n+2) : (2n+1))$ .
  - 5: **end for**
  - 6: Update  $\mathbf{a}$ :  $\mathbf{a} \leftarrow \mathbf{a} - t\mathbf{J}_{\tilde{\mathbf{c}}}\tilde{\mathbf{c}}$ , where  $t$  is the step size chosen such that the norm of the residual  $\tilde{\mathbf{c}}$  at the current step is smaller than the value at the previous step.
- 

### B.4 Implementation of the Newton step in the VF iteration employing the normal equations

---

**Algorithm 4**  $\mathbf{a} = \text{NEWTONonVF\_Normal}(\mathbf{H}, \omega, \mathbf{a})$

---

**Given:**  $\mathbf{H}, \omega, \mathbf{a}$  where  $H_i$  is the measured frequency response at  $f_i = \frac{\omega_i}{2\pi}$ , and  $\mathbf{a}$ , the initial set of starting poles.

**Output:**  $\mathbf{a}$ , the set of starting poles at the final step.

- 1: Solve  $(\mathbf{A}^*\mathbf{A})\mathbf{x} = \mathbf{A}^*\mathbf{b}$  for  $\mathbf{x}$ .
  - 2: **for**  $i = 1, \dots, n$  **do**
  - 3:     Compute  $\frac{\partial \mathbf{A}}{\partial a_i}$  using Eq.(6.45).
  - 4:     Compute  $RHS(:, i) = \left(\frac{\partial \mathbf{A}}{\partial a_i}\right)^* \mathbf{b} - \left(\left(\frac{\partial \mathbf{A}}{\partial a_i}\right)^* \mathbf{A} + \mathbf{A}^* \frac{\partial \mathbf{A}}{\partial a_i}\right) \mathbf{x}$ .
  - 5: **end for**
  - 6: Solve  $(\mathbf{A}^*\mathbf{A})\mathbf{J} = RHS$  for  $\mathbf{J}$  and set  $\mathbf{J}_{\tilde{\mathbf{c}}} = \mathbf{J}((n+2) : (2n+1), :)$ .
  - 7: Update  $\mathbf{a}$ :  $\mathbf{a} \leftarrow \mathbf{a} - t\mathbf{J}_{\tilde{\mathbf{c}}}\tilde{\mathbf{c}}$ , where  $\tilde{\mathbf{c}} = \mathbf{x}((n+2) : (2n+1))$  and  $t$  is the step size chosen such that the norm of the residual  $\tilde{\mathbf{c}}$  at the current step is smaller than the value at the previous step.
-

## B.5 Implementation of the Newton step in the VF iteration employing the pseudoinverse

---

**Algorithm 5**  $\mathbf{a} = \text{NEWTONonVF\_pinv}(\mathbf{H}, \omega, \mathbf{a})$

---

**Given:**  $\mathbf{H}$ ,  $\omega$ ,  $\mathbf{a}$  where  $H_i$  is the measured frequency response at  $f_i = \frac{\omega_i}{2\pi}$ , and  $\mathbf{a}$ , the initial set of starting poles.

**Output:**  $\mathbf{a}$ , the set of starting poles at the final step.

- 1: Find  $\mathbf{x}$  using  $\mathbf{x} = \mathbf{A}^+\mathbf{b}$ .
  - 2: **for**  $i = 1, \dots, n$  **do**
  - 3:   Compute  $\frac{\partial \mathbf{A}}{\partial a_i}$  using Eq.(6.45).
  - 4:   Compute  $\frac{\partial \mathbf{x}}{\partial a_i}$  using Eq.(6.50). Set  $\mathbf{J}_{\tilde{\mathbf{c}}}(:, i)$  as  $\frac{\partial \mathbf{x}}{\partial a_i}((n+2) : (2n+1))$ .
  - 5: **end for**
  - 6: Update  $\mathbf{a}$ :  $\mathbf{a} \leftarrow \mathbf{a} - t\mathbf{J}_{\tilde{\mathbf{c}}}\tilde{\mathbf{c}}$ , where  $\tilde{\mathbf{c}} = \mathbf{x}((n+2) : (2n+1))$  and  $t$  is the step size chosen such that the norm of the residual  $\tilde{\mathbf{c}}$  at the current step is smaller than the value at the previous step.
- 

## B.6 Obtaining real quantities in the least squares problem in vector fitting

To obtain real quantities in the least squares problem set up in Eq.(6.5) - (6.7), we assume that if a starting pole  $a_i$  is complex, then  $a_{i+1}$  is the complex conjugate of  $a_i$ , namely  $\bar{a}_i$ . Thus, the corresponding residues,  $c_i$  and  $c_{i+1}$ , as well as  $\tilde{c}_i$  and  $\tilde{c}_{i+1}$  share the same complex conjugate property. Moreover, since measurements are provided on the imaginary axis, we assume that, if  $s_i = j2\pi f_i$  and the corresponding measurement is  $H_1$ , then  $s_{i+1} = -j2\pi f_i$  with an associated measurement  $\bar{H}_1$ . Suppose, for simplicity, that we are trying to identify an order 3 system using one starting pole which is real,  $a_1$ , and two which are complex conjugates of each other,  $a_2$  and  $\bar{a}_2$ .

This leads to the following linear system:

$$\underbrace{\begin{bmatrix} \frac{1}{s_1-a_1} & \frac{1}{s_1-a_2} & \frac{1}{s_1-\bar{a}_2} & 1 & \frac{H_1}{s_1-a_1} & \frac{H_1}{s_1-a_2} & \frac{H_1}{s_1-\bar{a}_2} \\ \frac{1}{-s_1-a_1} & \frac{1}{-s_1-a_2} & \frac{1}{-s_1-\bar{a}_2} & 1 & \frac{\bar{H}_1}{-s_1-a_1} & \frac{\bar{H}_1}{-s_1-a_2} & \frac{\bar{H}_1}{-s_1-\bar{a}_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{s_N-a_1} & \frac{1}{s_N-a_2} & \frac{1}{s_N-\bar{a}_2} & 1 & \frac{H_N}{s_N-a_1} & \frac{H_N}{s_N-a_2} & \frac{H_N}{s_N-\bar{a}_2} \\ \frac{1}{-s_N-a_1} & \frac{1}{-s_N-a_2} & \frac{1}{-s_N-\bar{a}_2} & 1 & \frac{\bar{H}_N}{-s_N-a_1} & \frac{\bar{H}_N}{-s_N-a_2} & \frac{\bar{H}_N}{-s_N-\bar{a}_2} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \bar{c}_2 \\ d \\ \tilde{c}_1 \\ \tilde{c}_2 \\ \bar{\tilde{c}}_2 \end{bmatrix}}_{\mathbf{x}} \approx \underbrace{\begin{bmatrix} H_1 \\ \bar{H}_1 \\ \vdots \\ H_N \\ \bar{H}_N \end{bmatrix}}_{\mathbf{b}}. \quad (\text{B.1})$$

Post-multiplying  $\mathbf{A}$  by  $\Pi_a$  and pre-multiplying  $\mathbf{x}$  by  $\Pi_a^{-1}$ , where  $\Pi_a = \text{blkdiag}(1, \Pi, 1, 1, \Pi)$ ,

with  $\Pi = \begin{bmatrix} \frac{1}{2} & j \\ \frac{1}{2} & -j \end{bmatrix}$ , leads to  $\mathbf{A}_n = \mathbf{A}\Pi_a$ :

$$\mathbf{A}_n = \begin{bmatrix} \frac{1}{s_1-a_1} & \frac{1}{2} \left( \frac{1}{s_1-a_2} + \frac{1}{s_1-\bar{a}_2} \right) & \frac{j}{2} \left( \frac{1}{s_1-a_2} - \frac{1}{s_1-\bar{a}_2} \right) & 1 & \frac{H_1}{s_1-a_1} & \frac{1}{2} \left( \frac{H_1}{s_1-a_2} + \frac{H_1}{s_1-\bar{a}_2} \right) & \frac{j}{2} \left( \frac{H_1}{s_1-a_2} - \frac{H_1}{s_1-\bar{a}_2} \right) \\ \frac{1}{-s_1-a_1} & \frac{1}{2} \left( \frac{1}{-s_1-a_2} + \frac{1}{-s_1-\bar{a}_2} \right) & \frac{j}{2} \left( \frac{1}{-s_1-a_2} - \frac{1}{-s_1-\bar{a}_2} \right) & 1 & \frac{\bar{H}_1}{-s_1-a_1} & \frac{1}{2} \left( \frac{\bar{H}_1}{-s_1-a_2} + \frac{\bar{H}_1}{-s_1-\bar{a}_2} \right) & \frac{j}{2} \left( \frac{\bar{H}_1}{-s_1-a_2} - \frac{\bar{H}_1}{-s_1-\bar{a}_2} \right) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{s_N-a_1} & \frac{1}{2} \left( \frac{1}{s_N-a_2} + \frac{1}{s_N-\bar{a}_2} \right) & \frac{j}{2} \left( \frac{1}{s_N-a_2} - \frac{1}{s_N-\bar{a}_2} \right) & 1 & \frac{H_N}{s_N-a_1} & \frac{1}{2} \left( \frac{H_N}{s_N-a_2} + \frac{H_N}{s_N-\bar{a}_2} \right) & \frac{j}{2} \left( \frac{H_N}{s_N-a_2} - \frac{H_N}{s_N-\bar{a}_2} \right) \\ \frac{1}{-s_N-a_1} & \frac{1}{2} \left( \frac{1}{-s_N-a_2} + \frac{1}{-s_N-\bar{a}_2} \right) & \frac{j}{2} \left( \frac{1}{-s_N-a_2} - \frac{1}{-s_N-\bar{a}_2} \right) & 1 & \frac{\bar{H}_N}{-s_N-a_1} & \frac{1}{2} \left( \frac{\bar{H}_N}{-s_N-a_2} + \frac{\bar{H}_N}{-s_N-\bar{a}_2} \right) & \frac{j}{2} \left( \frac{\bar{H}_N}{-s_N-a_2} - \frac{\bar{H}_N}{-s_N-\bar{a}_2} \right) \end{bmatrix} \quad (\text{B.2})$$

and  $\mathbf{x}_n = \Pi_a^{-1} \mathbf{x} = \begin{bmatrix} c_1 & c_2 + \bar{c}_2 & -j(c_2 - \bar{c}_2) & d & \tilde{c}_1 & \tilde{c}_2 + \bar{\tilde{c}}_2 & -j(\tilde{c}_2 - \bar{\tilde{c}}_2) \end{bmatrix}^T$ . We simplify these expressions in terms of the real and imaginary parts of the quantities involved:

$$\mathbf{A}_n = \begin{bmatrix} \frac{1}{s_1-a_1} & \frac{s_1-\Re(a_2)}{(s_1-\Re(a_2))^2+\Im(a_2)^2} & \frac{-\Im(a_2)}{(s_1-\Re(a_2))^2+\Im(a_2)^2} & 1 & \frac{H_1}{s_1-a_1} & \frac{H_1(s_1-\Re(a_2))}{(s_1-\Re(a_2))^2+\Im(a_2)^2} & \frac{-H_1\Im(a_2)}{(s_1-\Re(a_2))^2+\Im(a_2)^2} \\ \frac{-1}{s_1+a_1} & \frac{-s_1-\Re(a_2)}{(s_1+\Re(a_2))^2+\Im(a_2)^2} & \frac{-\Im(a_2)}{(s_1+\Re(a_2))^2+\Im(a_2)^2} & 1 & \frac{-\bar{H}_1}{s_1+a_1} & \frac{-\bar{H}_1(s_1+\Re(a_2))}{(s_1+\Re(a_2))^2+\Im(a_2)^2} & \frac{-\bar{H}_1\Im(a_2)}{(s_1+\Re(a_2))^2+\Im(a_2)^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{s_N-a_1} & \frac{(s_N-\Re(a_2))}{(s_N-\Re(a_2))^2+\Im(a_2)^2} & \frac{-\Im(a_2)}{(s_N-\Re(a_2))^2+\Im(a_2)^2} & 1 & \frac{H_N}{s_N-a_1} & \frac{H_N(s_N-\Re(a_2))}{(s_N-\Re(a_2))^2+\Im(a_2)^2} & \frac{-H_N\Im(a_2)}{(s_N-\Re(a_2))^2+\Im(a_2)^2} \\ \frac{-1}{s_N+a_1} & \frac{-s_N-\Re(a_2)}{(s_N+\Re(a_2))^2+\Im(a_2)^2} & \frac{-\Im(a_2)}{(s_N+\Re(a_2))^2+\Im(a_2)^2} & 1 & \frac{-\bar{H}_N}{s_N+a_1} & \frac{-\bar{H}_N(s_N+\Re(a_2))}{(s_N+\Re(a_2))^2+\Im(a_2)^2} & \frac{-\bar{H}_N\Im(a_2)}{(s_N+\Re(a_2))^2+\Im(a_2)^2} \end{bmatrix} \quad (\text{B.3})$$



and  $\mathbf{x}_n^T = \left[ c_1 \quad 2\Re(c_2) \quad 2\Im(c_2) \quad d \quad \tilde{c}_1 \quad 2\Re(\tilde{c}_2) \quad 2\Im(\tilde{c}_2) \right]$ . To obtain real quantities in terms of the measurements, we pre-multiply both  $\mathbf{A}_n$  and  $\mathbf{b}$  by  $\Pi_s = \text{blkdiag}(\Pi, \dots, \Pi)^*$ . This leads to  $\mathbf{A}_r = \Pi_s^* \mathbf{A}_n = \Pi^* \mathbf{A} \Pi_a$  and  $\mathbf{b}_r = \Pi_s^* \mathbf{b}$ :

$$\mathbf{A}_r = \begin{bmatrix} \Re\left(\frac{1}{s_1 - a_1}\right) & \Re\left(\frac{(s_1 - \Re(a_2))}{(s_1 - \Re(a_2))^2 + \Im(a_2)^2}\right) & \dots \\ \Im\left(\frac{1}{s_1 - a_1}\right) & \Im\left(\frac{(s_1 - \Re(a_2))}{(s_1 - \Re(a_2))^2 + \Im(a_2)^2}\right) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad \mathbf{b}_r = \begin{bmatrix} \Re(H_1) \\ \Im(H_1) \\ \vdots \end{bmatrix} \quad (\text{B.4})$$

Note that the pre-multiplication by  $\Pi_s$  does not affect the solution  $\mathbf{x}_r$ . Therefore, we now have a least squares problem with real quantities to solve:  $\mathbf{A}_r \mathbf{x}_r \approx \mathbf{b}_r$ .

The next step is to make  $\frac{\partial \mathbf{A}}{\partial a_i}$  real. First, we discuss the case in which  $a_i$  is a real starting pole, so the only non-zero quantity of  $\frac{\partial \Phi}{\partial a_i}$  is its  $i^{\text{th}}$  column:

$$\frac{\partial \Phi}{\partial a_i} = \begin{bmatrix} 0 & \dots & \frac{1}{(s_1 - a_i)^2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \frac{1}{(s_N - a_i)^2} & \dots & 0 \end{bmatrix}. \quad (\text{B.5})$$

If  $a_i$  is complex, then we need to take the derivative with respect to its real and imaginary parts. Assuming that  $a_{i+1}$  is the complex conjugate of  $a_i$ , we have that

$$\frac{\partial \Phi}{\partial \Re(a_i)} = \begin{bmatrix} 0 & \dots & \frac{(s_1 - \Re(a_i))^2 - \Im(a_i)^2}{[(s_1 - \Re(a_i))^2 + \Im(a_i)^2]^2} & \frac{-2(s_1 - \Re(a_i))\Im(a_i)}{[(s_1 - \Re(a_i))^2 + \Im(a_i)^2]^2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \frac{(s_N - \Re(a_i))^2 - \Im(a_i)^2}{[(s_N - \Re(a_i))^2 + \Im(a_i)^2]^2} & \frac{-2(s_N - \Re(a_i))\Im(a_i)}{[(s_N - \Re(a_i))^2 + \Im(a_i)^2]^2} & \dots & 0 \end{bmatrix} \quad (\text{B.6})$$

and

$$\frac{\partial \Phi}{\partial \Im(a_i)} = \begin{bmatrix} 0 & \dots & \frac{-2(s_1 - \Re(a_i))\Im(a_i)}{[(s_1 - \Re(a_i))^2 + \Im(a_i)^2]^2} & \frac{-(s_1 - \Re(a_i))^2 + \Im(a_i)^2}{[(s_1 - \Re(a_i))^2 + \Im(a_i)^2]^2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \frac{-2(s_N - \Re(a_i))\Im(a_i)}{[(s_N - \Re(a_i))^2 + \Im(a_i)^2]^2} & \frac{-(s_N - \Re(a_i))^2 + \Im(a_i)^2}{[(s_N - \Re(a_i))^2 + \Im(a_i)^2]^2} & \dots & 0 \end{bmatrix}. \quad (\text{B.7})$$

To obtain real quantities in terms of the measurements, we pre-multiply  $\frac{\partial \mathbf{A}}{\partial a_i}$  by  $\Pi_s$ .

It is straightforward to implement these changes in Algorithms 3 - 5.

## Additional numerical examples in the Loewner framework

---

### C.1 200 S-parameter measurements from a device with 26 ports

This set contains  $N = 200$  S-parameter measurements for frequency samples between 5MHz and 1GHz. For better conditioning, all frequencies were scaled by  $10^{-6}$ .

Fig. C.1(a) shows the normalized singular values of the Loewner and shifted Loewner matrices constructed using all samples, in the complex and real approaches. We notice the same behavior as before. The singular values of the Loewner matrices show a decay between the 48th and 49th, while for the shifted Loewner matrices, it is between the 74th and 75th. The singular value drop suggests that there is an underlying **D**-term and, moreover, allows to identify the system order.

Table C.1 presents results for stable models obtained with our approaches. VF was required to produce a **D** matrix, but the lowest order model we could construct was  $n=52$ . All proposed algorithms yield better fits than VF.

For comparable errors, VF needs to build a model of order  $n=780$ . Fig. C.1(b)

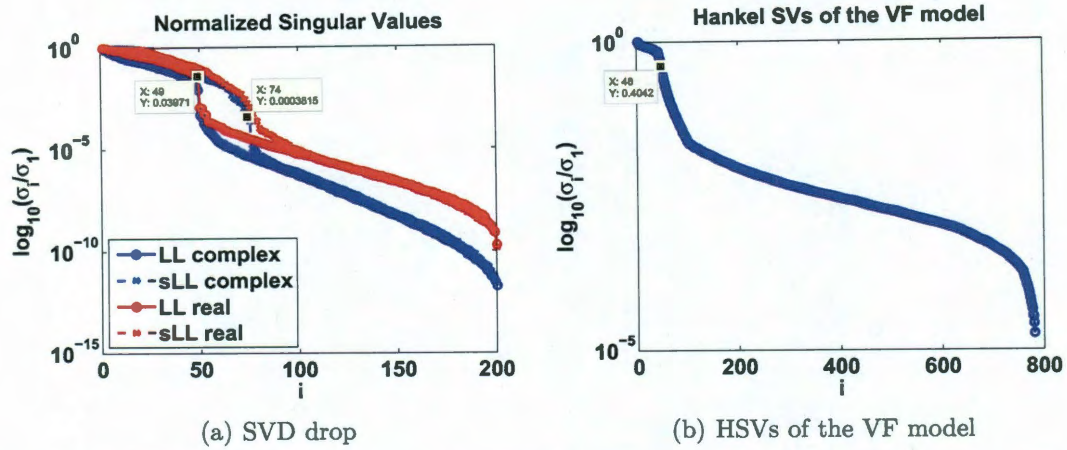


Figure C.1: Singular value drop of the Loewner matrix pencil and Hankel singular value drop of the VF model

Algorithm	CPU time (s)	$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
<b>VF (n=52)</b>	<b>1.12</b>	<b>1.1554</b>	<b>2.9983e-001</b>
SVD Complex (n=48)	0.39	1.1073e-001	1.0167e-002
<b>SVD Real (n=48)</b>	<b>0.42</b>	<b>5.7958e-002</b>	<b>6.5341e-003</b>
Adaptive Complex (n=48)	1.74	1.7622e-001	2.2432e-002
Adaptive Real (n=48)	1.77	7.7407e-002	1.0332e-002
VF (n=780)	18.17	3.8320e-002	5.8102e-003
VF & BT (n=48)	40.87	5.3897e-001	2.0251e-001

Table C.1: Results for a device with  $p = 26$  ports

shows a slow decay in the Hankel singular values of this VF model, so the balanced truncation (BT) reduced model of order  $n=48$  is poor, as seen from Table C.1.

The model obtained with the real SVD approach is shown in Fig. C.2(a), while the VF model of order 52 is shown in Fig. C.2(b). The plots in Fig. C.3(a) and C.3(b) show the singular values of the error matrices.

Fig. C.4 compares the measured  $S_{1,2}$  and  $S_{1,26}$  entries to the model obtained with the SVD approach and with VF.

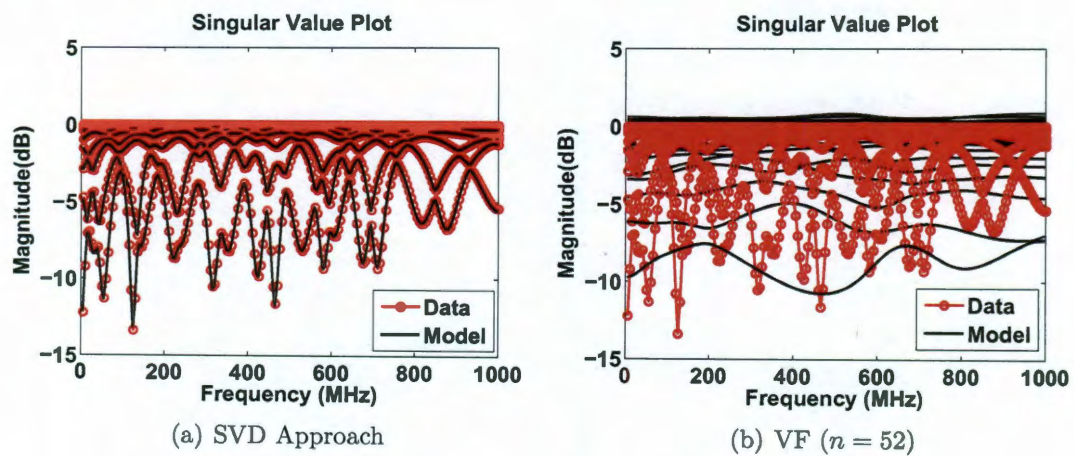


Figure C.2: Models for a device with  $p = 26$  ports

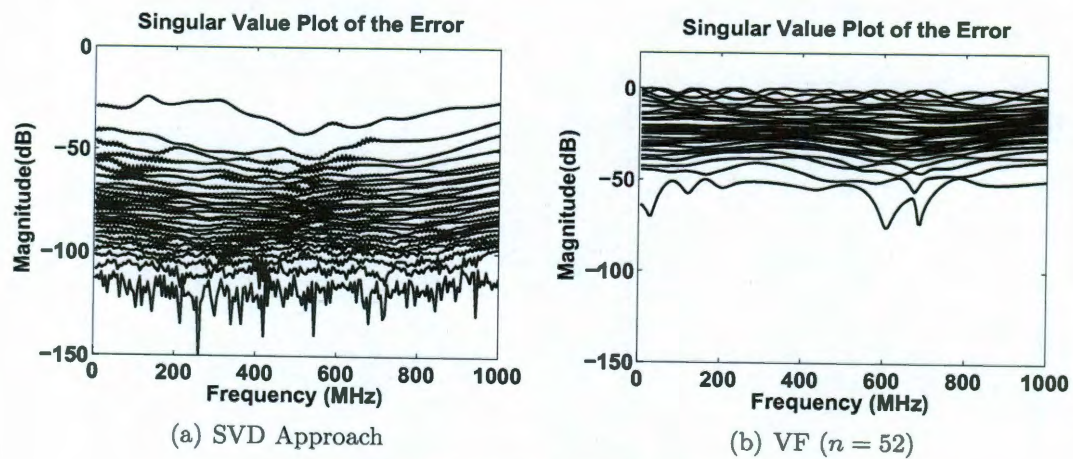


Figure C.3: Error plots for a device with  $p = 26$  ports

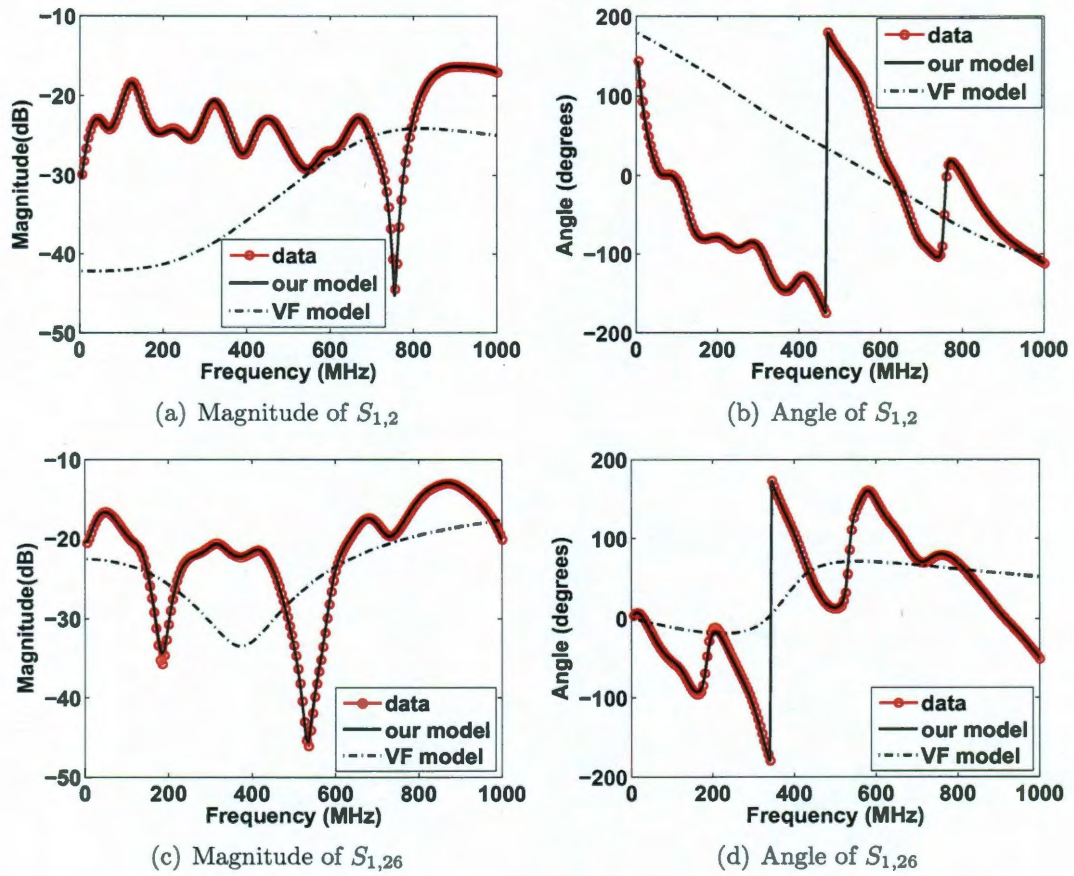


Figure C.4: Modeling S-parameter entries for a device with  $p = 26$  ports

## C.2 Delay System

Electrical circuits with transmission lines can be described by differential equations containing delays. These expressions are obtained from the d'Alembert solution to the telegrapher's equations [81]. The delay factor turns out to be  $\sqrt{LC}$ .

In the following, we consider the system:

$$\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}_0\mathbf{x}(t) + \mathbf{A}_1\mathbf{x}(t - \tau) + \mathbf{b}u(t), \quad \mathbf{y}(t) = \mathbf{c}\mathbf{x}(t), \quad (\text{C.1})$$

with  $\mathbf{E}$ ,  $\mathbf{A}_0$ ,  $\mathbf{A}_1 \in \mathbb{R}^{500 \times 500}$  and  $\tau = 1$ . The associated transfer function is given by

$$\mathbf{H}(s) = \mathbf{c} (\mathbf{sE} - \mathbf{A}_0 - \mathbf{A}_1 e^{-\tau s})^{-1} \mathbf{b}. \quad (\text{C.2})$$

The matrices  $\mathbf{E}$ ,  $\mathbf{A}_0$  and  $\mathbf{A}_1$  are symmetric [50]. Moreover,  $\mathbf{E}$  is positive definite, while  $\mathbf{A}_0$  and  $\mathbf{A}_1$  are negative definite. Due to the time delay, this is an infinite-dimensional system. However, as the matrices  $\mathbf{E}$ ,  $\mathbf{A}_0$  and  $\mathbf{A}_1$  are diagonalized by the same transformation, the system can be decoupled into 500 independent equations and the poles of the system can be found by solving nonlinear equations of the type:

$$s\lambda_{\mathbf{E}} - \lambda_{\mathbf{A}_0} - \lambda_{\mathbf{A}_1} e^{-\tau s} = 0, \quad (\text{C.3})$$

where  $\lambda_{\mathbf{E}}$  is an eigenvalue of  $\mathbf{E}$ , and  $\lambda_{\mathbf{A}_0}$  and  $\lambda_{\mathbf{A}_1}$  are eigenvalues of  $\mathbf{A}_0$  and  $\mathbf{A}_1$ , respectively. Setting  $s$  as  $x + j \cdot y$  allows us to write Eq. (C.3) as a system of 2 nonlinear equations corresponding to the real and imaginary part:

$$x\lambda_{\mathbf{E}} - \lambda_{\mathbf{A}_0} - \lambda_{\mathbf{A}_1} \cos(\tau x) = 0 \quad \text{and} \quad y\lambda_{\mathbf{E}} + \lambda_{\mathbf{A}_1} \sin(\tau y) = 0. \quad (\text{C.4})$$

Figure C.5(a) shows poles with imaginary part between  $\pm 130$ . Each of the 500 systems of equations of the form (C.4) were solved using MATLAB's `fsolve` with initial guess 0 for  $x$  and different guesses for  $y$ , between  $-130$  and  $130$ . Thus, for each value on the imaginary axis, there is one pole obtained from solving one of the systems of

nonlinear equations. Even though Fig. C.5(a) shows only poles with imaginary part between  $-130$  and  $130$ , they extend well above that value.

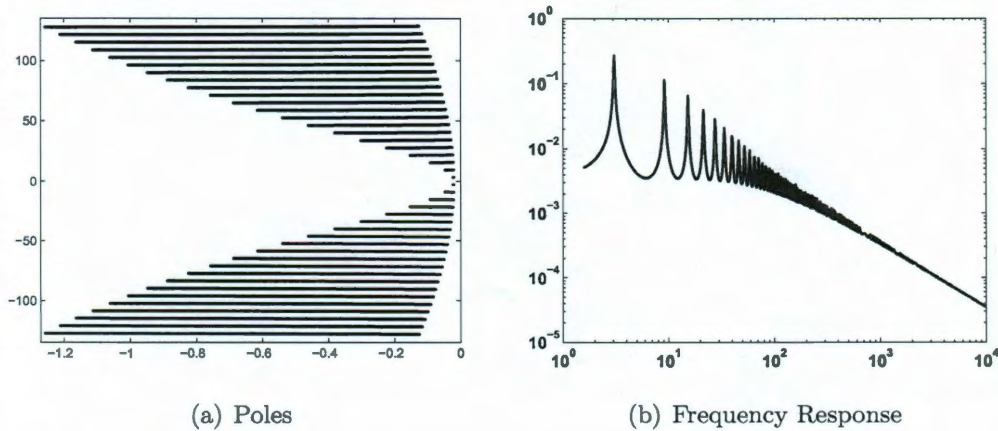


Figure C.5: Poles and frequency response of the delay system

We are looking for a low-order finite-dimensional model as in Eq. (3.1) to approximate this infinite-dimensional system. We consider 1000 measurements of the frequency response logarithmically distributed between  $10^{-0.6}$  and  $10^{3.2}$  and use this information together with the theoretical concepts described in Sect. 4.1.3 and the numerical implementation presented in Sect. 4.2 to construct such a reduced model. We employ state-of-the-art *vector fitting* (see Chapter 6) in our comparison.

Let us begin our discussion with the analysis of the singular value drop of the matrix pencil  $(\sigma\mathbb{L}, \mathbb{L})$  constructed using all measurements. This is shown in Figure C.6. We notice that the rank of the pencil  $(\sigma\mathbb{L}, \mathbb{L})$  constructed using these measurements is about 700. Since the system is infinite-dimensional, one may expect the matrix pencil to be full rank. The fact that it has a numerical rank of 700, instead of 1000, is because of the small magnitude of the response and small oscillation peaks at high frequencies, which can essentially be approximated by a line (see Fig. C.5(b)).

We build models of order  $n = 41$ . Results of the comparison of our approach against vector fitting are shown in Table C.2, while Fig. C.7 presents plots of the

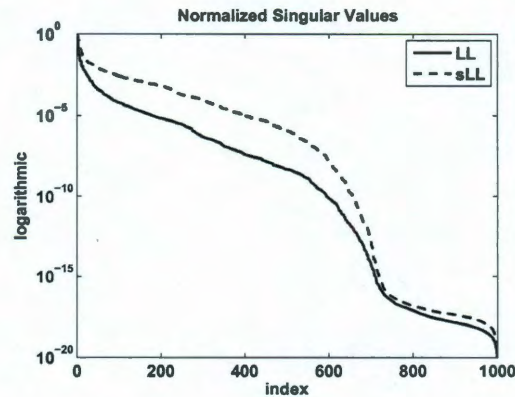


Figure C.6: Singular value drop of the Loewner matrix pencil

errors in absolute value between the original system and the different reduced models.

Algorithm	Iterations	$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
SVD Approach	N/A	1.4995e-002	3.5498e-002
Vector Fitting	50	3.4159e-002	5.4146e-002

Table C.2: Results for models of order  $n = 41$  for an infinite-dimensional delay system

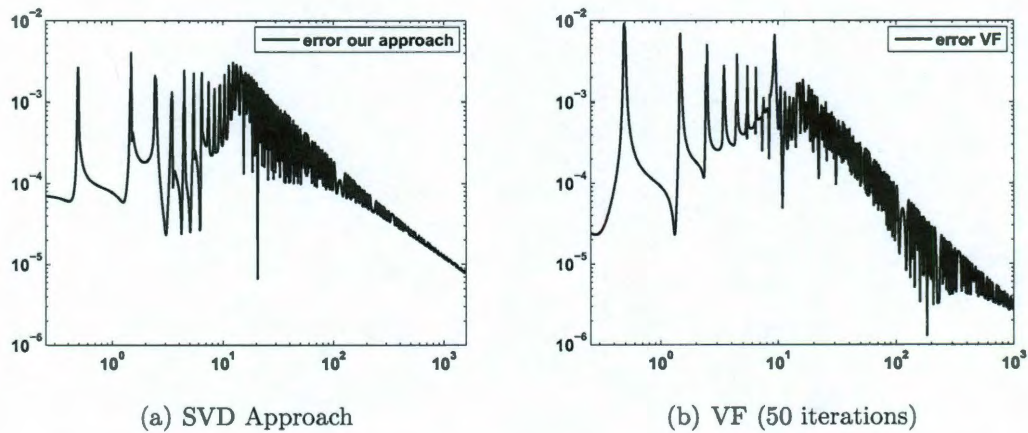


Figure C.7: Models obtained with our approach and with vector fitting

Figure C.8(a) compares the  $\mathcal{H}_\infty$  error between all our models with orders between  $n = 1$  and  $n = 50$  and the same order model built with vector fitting (in 50 iterations). Similarly, Fig. C.8(b) shows the corresponding  $\mathcal{H}_2$  errors. We notice that, in most cases, the errors for our models are lower than the VF models. Moreover, in Fig.



C.8(a), we observe that VF reaches a plateau, in the sense that the errors stay almost constant with increasing the model order. The opposite behaviour happens for our models, as they become better when we increase the desired dimension.

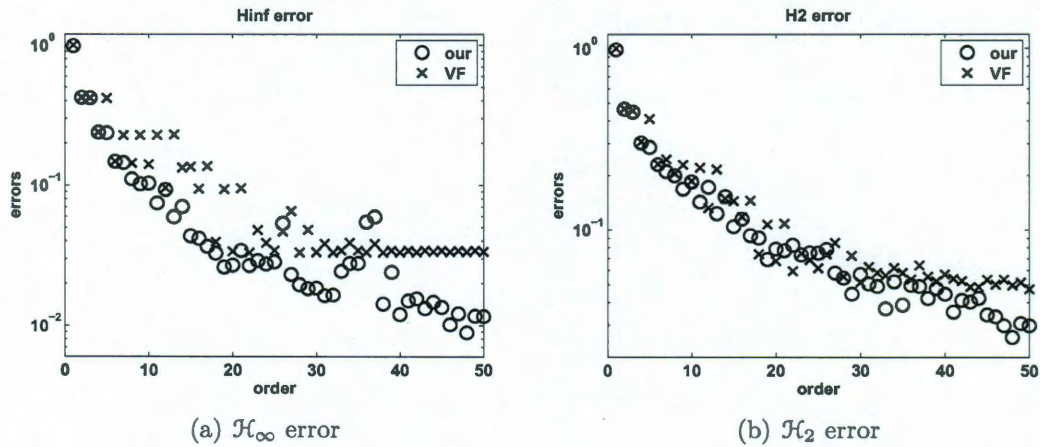


Figure C.8:  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$  errors for our approach compared to vector fitting

Since vector fitting solves an optimization problem, its performance greatly depends on the number of iterations. Fig. C.9 shows the decay of the  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$  errors as a function of the number of iterations performed (between 1 and 50) when the procedure uses 41 starting poles. The errors stagnate after about 20 iterations.

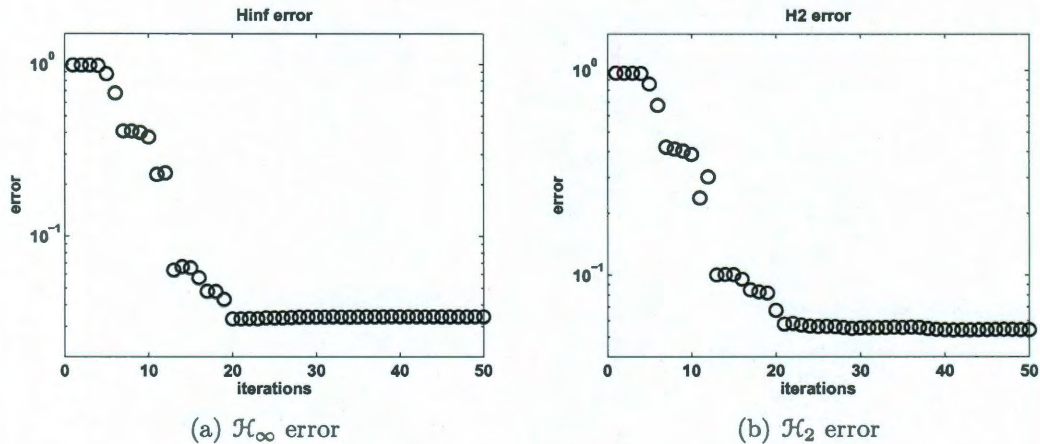


Figure C.9:  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$  errors for VF as a function of the number of iterations

### C.3 Circuit with $p = 70$ ports

We consider a dynamical system which describes a linear subcircuit of a large post-layout circuit (i.e. with parasitics due to layout). It was provided by Qimonda AG under the name *rc549* and has previously been investigated in [82] from the point of view of terminal reduction [50], an approach used for model reduction of systems with a large number of terminals [83, 84]. This system has 141 poles: 84 infinite, 56 real between  $-1.9 \cdot 10^{15}$  and  $-1.1 \cdot 10^{13}$  and one at  $-2 \cdot 10^{-1}$ . The poles are depicted in Fig. C.10(b) and the frequency response of this system is shown in Fig. C.10(a).

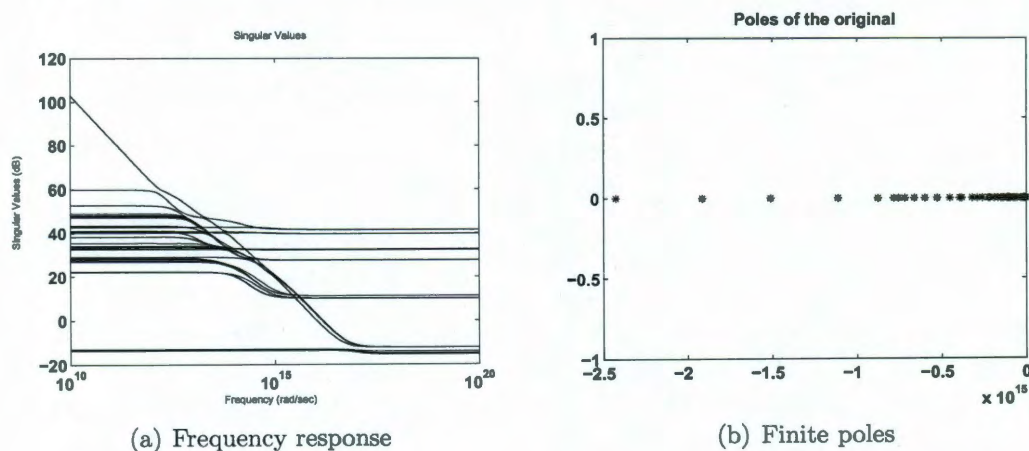
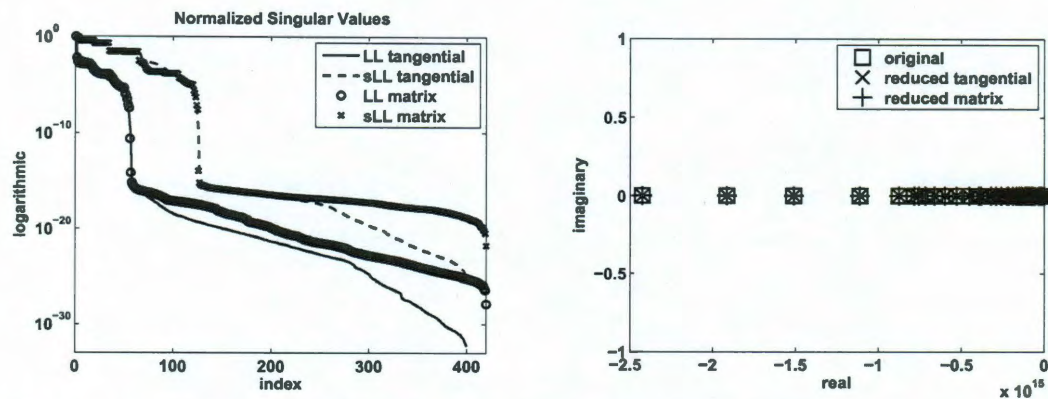


Figure C.10: Frequency response and finite poles of the original system

Our goal is to compare the tangential approach to the matrix interpolation approach on this example arising in circuit simulation. Authors in [85] suggested to use matrix interpolation instead of tangential interpolation, as the resulting model is of the same quality, but the number of measurements needed is much smaller, especially when the number of ports is large. For tangential interpolation, we take  $N = 400$  measurements of the transfer function logarithmically distributed between  $10^9$  and  $10^{20}$  and, for each measurement, we select one sampling direction as one of the unit vectors of dimension  $p$ . Thus, for each frequency measurement, we use either a column (as right data) or a row (as left data) of the matrix measurement. The



(a) Drop of the singular values of the Loewner matrix pencil for tangential and matrix interpolation (b) Poles of the original and reduced systems

Figure C.11: Singular values of the Loewner matrix pencil and poles of the original and reduced systems

resulting Loewner and shifted Loewner matrices are of dimension 400 and a plot of their normalized singular values is shown in Fig. C.11(a). For matrix interpolation, we take  $N = 6$  measurements of the transfer function logarithmically distributed between  $10^9$  and  $10^{20}$  and, for each measurement, we select  $p$  sampling directions as  $p$  unit vectors (all columns and rows of the identity matrix of dimension  $p \times p$  are used as right directions and left directions, respectively). Thus, for each frequency measurement, we use the entire  $p \times p$  matrix measurement. The resulting Loewner and shifted Loewner matrices are of dimension  $6 \times 70 = 420$  and a plot of their normalized singular values is also shown in Fig. C.11(a). We notice that both methods yield Loewner and shifted Loewner matrices of the same rank, which suggest that the underlying system can be reduced to order  $n = 127$ .

Figure C.11(b) shows the finite poles of the original system, together with the poles of the reduced systems obtained with the tangential and the matrix interpolation approaches. We notice that all poles were recovered. In particular, we notice that the reduced systems of order  $n = 127$  with  $\mathbf{D} = \mathbf{0}$  contain 57 finite poles (the same as the original one), and 70 infinite poles which hide an underlying non-zero  $\mathbf{D}$ -term,

as in Eq. (4.36). Hence, all redundant poles have been eliminated.

Table C.3 shows that both approaches yield small errors, with the tangential interpolation approach yielding slightly smaller errors than matrix interpolation.

Approach	$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
Tangential Interpolation	3.3422e-010	3.3589e-010
Matrix Interpolation	6.9040e-009	6.9056e-009

Table C.3: Results for the tangential and matrix interpolation approaches

Figure C.12 shows the frequency response (sigma plot) of the top left and bottom right  $10 \times 10$  entries of the original system, together with the reduced systems obtained via the two approaches: tangential and matrix interpolation.

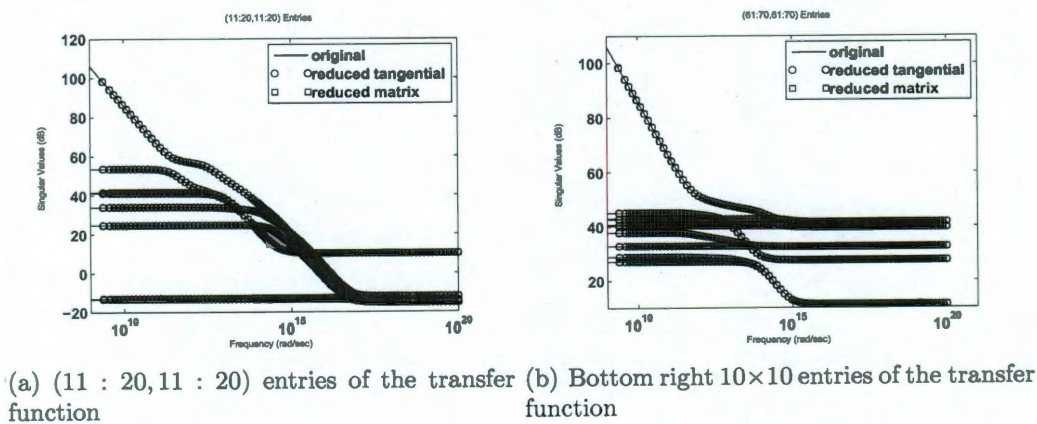


Figure C.12: Entries of the original transfer function plotted against those of the reduced systems

The purpose of this example is to show that tangential interpolation yields comparable results to matrix interpolation provided that the sampling directions are chosen as outlined in Sect. 4.1.4. Moreover, both methods are able to eliminate the redundancy and produce reduced models which are very close to the original.

## C.4 Example from Sect. C.1 revisited

S-parameter data were provided in magnitude-angle format (with at most 9 significant digits for the magnitude and at most 6 significant digits for the angle). This data set contains  $k = 200$  samples between 5MHz and 1GHz of a device with 26 ports [50].

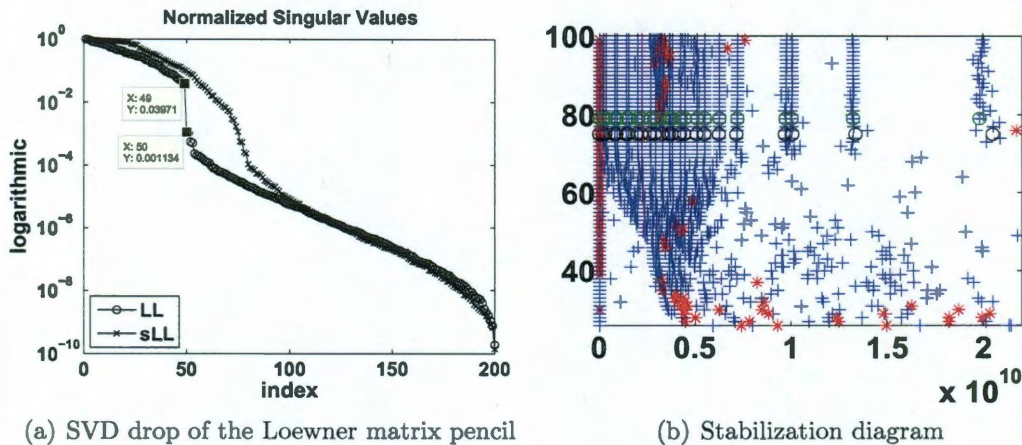


Figure C.13: Plots for the data set obtained from a device with 26 ports

We build the Loewner matrix pencil using the real alternative approach (thus, the results are slightly smaller than those presented in Sect. C.1). The singular value drop of  $x\mathbb{L} - \sigma\mathbb{L}$  (shown in Fig. C.13(a)), where  $x = 2\pi f_1$ , reveals a drop in the singular values of the Loewner matrices between indices 49 and 50, but the drop in the singular values of the shifted Loewner matrix is not very clear. Thus, we believe the order of the underlying system is 49, together with a full rank  $\mathbf{D}$ -term, leading to a realization of order  $49 + 26 = 75$ .

We proceed with building the stabilization diagram (see Fig. C.13(b)). This shows that, by order 75, all poles are approximated, but as the order is increased, the estimates start converging. The order 79 model is the lowest order model which provides good approximations of the poles, so we apply the largest residues and largest dominance criteria to trim the 75 physical poles from the rest. Both indicate that the underlying system is of order 75 (Table C.4). The first 26 poles, some of which

are unstable, correspond to poles at infinity. The black circles in Fig. C.13(b) are the poles at order 75, while the green circles are the 75 most dominant poles of the order 79 system. The 75 dominant poles obtained from the order 79 model are better approximations of the true poles than the ones obtained from the order 75 model.

$i$	$\tilde{\lambda}_i$	$\ \mathbf{R}_i\ _2$	$q_i$
1	1.86e+13	1.02e+14	5.47e+0
2	5.58e+12	1.01e+14	1.80e+1
3-4	3.85e+12±1.74e+12i	4.60e+13	1.19e+1
⋮	⋮	⋮	⋮
72-73	-1.88e±8 +2.91e+9i	2.56e+8	1.35e+0
74-75	-1.65e+8±7.89e+8i	2.18e+8	1.32e+0
76-77	2.54e+7±3.89e+9i	1.34e+6	5.31e-2
78-79	-3.19e+6±3.40e+9i	9.68e+5	3.03e-1

Table C.4: Dominance quantities

Model	$\mathcal{H}_\infty$ error	$\mathcal{H}_2$ error
Order 75	1.95e-2	1.98e-3
Order 79	9.87e-3	8.75e-4
Dominant 75	7.25e-3	8.59e-4

Table C.5: Error norm results

Table C.5 shows the error norms of the various systems we considered. Clearly, the smallest errors are obtained by considering the order 75 model obtained by selecting the most dominant poles of the order 79 system.

We plot the response of the system obtained with the proposed procedure in Fig. C.14. It is hardly distinguishable from the data.

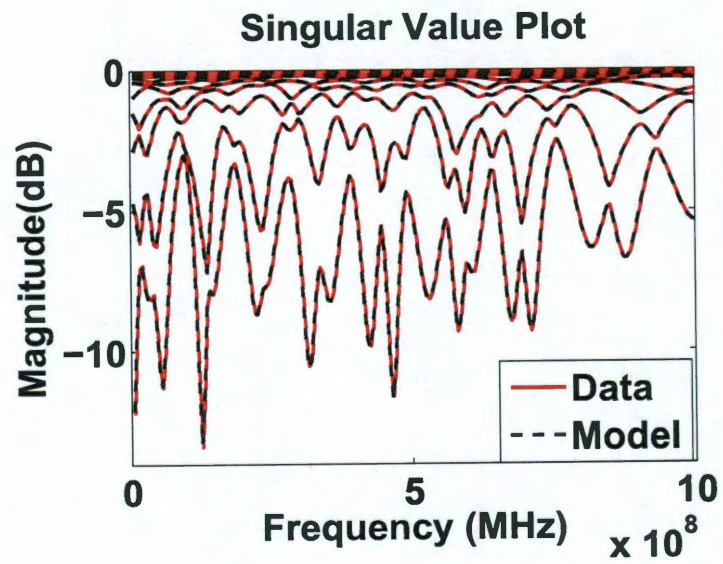


Figure C.14: Frequency response

## Additional numerical examples analyzing the convergence properties of vector fitting

---

In the following sections, we discuss additional numerical examples in which we investigate the convergence properties of the pole relocation iteration in vector fitting.

### D.1 Order 1 strictly proper system

The underlying system that we are modeling is of the form

$$H(s) = \frac{r}{s - p}. \quad (\text{D.1})$$

We use one starting pole  $a_1$ , so the expression we obtain is

$$f(s) = \frac{\frac{c_1}{s - a_1}}{\frac{\tilde{c}_1}{s - a_1} + 1} = \frac{c_1}{s - (a_1 - \tilde{c}_1)}. \quad (\text{D.2})$$

At the next iteration, the starting pole is given by the zero of the polynomial in the denominator, namely  $a_1 - \tilde{c}_1$ , where  $\tilde{c}_1$  is found by solving the least squares problem in Eq.(6.4).



### D.1.1 2 measurements

Suppose we have 2 measurements available of  $H(s)$  at  $s_1$  and  $s_2$ , namely  $H_1$  and  $H_2$ . In this case, we have two pieces of information given ( $H_1$  and  $H_2$ ) and two unknowns ( $a$  and  $b$ ), so  $a$  and  $b$  can be found in one step, without any iteration involved. The linear system we are solving is  $\mathbf{A}(s_i, 2; a_i, 1)\mathbf{x}(c_i, 1; \tilde{c}_i, 1) = \mathbf{b}$  with  $\mathbf{A}(s_i, 2; a_i, 1) \in \mathbb{C}^{2 \times 2}$ ,  $\mathbf{x}(c_i, 1; \tilde{c}_i, 1), \mathbf{b} \in \mathbb{C}^{2 \times 1}$ . The solution can be found exactly, since the system is not overdetermined and we are not solving a least squares problem:

$$\mathbf{x} = \begin{bmatrix} c_1 \\ \tilde{c}_1 \end{bmatrix} = \begin{bmatrix} \frac{H_1 H_2 (s_2 - s_1)}{H_1 - H_2} \\ \frac{H_2 (s_2 - a_1) - H_1 (s_1 - a_1)}{H_1 - H_2} \end{bmatrix}, \quad (\text{D.3})$$

so the new starting pole is  $a_1 - \tilde{c}_1 = \frac{H_1 s_1 - H_2 s_2}{H_1 - H_2}$ . Note that this quantity does not depend on the initial starting pole, but rather, only on measurements. This implies that there is no iteration, or that the iteration stagnates after the first step.

### D.1.2 $N \geq 2$ noise-free measurements

The underlying system is of the form  $H(s) = \frac{r}{s-p}$ , with  $r$ , the residue, and  $p$ , the pole, so  $H_1 = \frac{r}{s_1-p}$ ,  $H_2 = \frac{r}{s_2-p}, \dots, H_N = \frac{r}{s_N-p}$ . Therefore, the matrix  $\mathbf{A}$  is

$$\mathbf{A}(s_i, N; a_i, 1) = \begin{bmatrix} \frac{1}{s_1 - a_1} & -\frac{H_1}{s_1 - a_1} \\ \vdots & \vdots \\ \frac{1}{s_N - a_1} & -\frac{H_N}{s_N - a_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{s_1 - a_1} & -\frac{r}{(s_1 - p)(s_1 - a_1)} \\ \vdots & \vdots \\ \frac{1}{s_N - a_1} & -\frac{r}{(s_N - p)(s_N - a_1)} \end{bmatrix}, \quad (\text{D.4})$$

while the right hand side vector  $\mathbf{b}$  is

$$\mathbf{b} = \begin{bmatrix} H_1 \\ \vdots \\ H_N \end{bmatrix} = \begin{bmatrix} \frac{r}{s_1 - p} \\ \vdots \\ \frac{r}{s_N - p} \end{bmatrix}, \quad (\text{D.5})$$

so the least squares problem we are solving is

$$\underbrace{\begin{bmatrix} \frac{1}{s_1 - a_1} & -\frac{r}{(s_1 - p)(s_1 - a_1)} \\ \vdots & \vdots \\ \frac{1}{s_N - a_1} & -\frac{r}{(s_N - p)(s_N - a_1)} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_1 \\ \tilde{c}_1 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \frac{r}{s_1 - p} \\ \vdots \\ \frac{r}{s_N - p} \end{bmatrix}}_{\mathbf{b}}. \quad (\text{D.6})$$

By inspection, we recognize that the solution is

$$\begin{bmatrix} c_1 \\ \tilde{c}_1 \end{bmatrix} = \begin{bmatrix} r \\ a_1 - p \end{bmatrix} \quad (\text{D.7})$$

and one can verify this using any equation of the linear system:

$$\frac{c_1}{s_i - a_1} - \frac{r\tilde{c}_1}{(s_i - p)(s_i - a_1)} = \frac{r}{s_i - p} \quad (\text{D.8})$$

$$\Rightarrow \frac{r}{s_i - a_1} - \frac{r(a_1 - p)}{(s_i - p)(s_i - a_1)} = \frac{r}{s_i - p} \quad (\text{D.9})$$

$$\Leftrightarrow r \frac{(s_i - p) - (a_1 - p)}{(s_i - p)(s_i - a_1)} = \frac{r}{s_i - p} \quad (\text{D.10})$$

$$\Leftrightarrow r \frac{(s_i - a_1)}{(s_i - p)(s_i - a_1)} = \frac{r}{s_i - p}. \quad (\text{D.11})$$

The last equation indeed holds for any  $s_i$ , so the solution vector in Eq.(D.7) is the right one. Therefore, the model found by Vector Fitting is (see Eq.(D.2))

$$f(s) = \frac{c_1}{s - (a_1 - \tilde{c}_1)} = \frac{r}{s - (a_1 - a_1 + p)} = \frac{r}{s - p} = H(s) \quad (\text{D.12})$$

so the original system has been recovered. At the second step,  $a_1$  is taken as the zero of the denominator, which is  $p$ . Thus,  $\tilde{c}_1 = a_1 - p = 0$ , so convergence is reached at the second step.

### D.1.3 3 noisy measurements

The underlying system is of the form  $H(s) = \frac{r}{s-p}$ , with  $r$ , the residue, and  $p$ , the pole, but instead of noise-free measurements, we are provided with  $H_1 = \frac{r}{s_1-p} + n_1$ ,  $H_2 = \frac{r}{s_2-p} + n_2$  and  $H_3 = \frac{r}{s_3-p} + n_3$ , where  $n_i$ ,  $i = 1, \dots, 3$  are values of the measurement noise. At this point we make no assumptions on the distribution of the noise or its magnitude. Suppose the frequencies where measurements are provided are distinct, therefore the measurements available are also distinct:  $s_1 \neq s_2$ ,  $s_2 \neq s_3$ ,  $s_3 \neq s_1$  and, consequently,  $H_1 \neq H_2$ ,  $H_2 \neq H_3$ ,  $H_3 \neq H_1$ . The linear system we are solving is  $\mathbf{A}(s_i, N; a_i, 1)\mathbf{x}(c_i, 1; \tilde{c}_i, 1) = \mathbf{b}$ . The system is overdetermined, so the solution  $\mathbf{x}$  is found via least squares. The starting pole at the next iteration is simply  $a_1 - \tilde{c}_1$ . Since this analysis is geared towards finding the convergence points, we use the convergence condition that the starting pole stays constant to write  $a_1^{(k)} = a_1^{(k+1)}$ , yielding  $a_1^{(k)} = a_1^{(k)} - \tilde{c}_1^{(k)}$ , so  $\tilde{c}_1 = 0$ .

We first use the normal equations to write the solution  $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ , so

$$\tilde{c}_1 = \frac{\alpha_3 a_1^3 + \alpha_2 a_1^2 + \alpha_1 a_1 + \alpha_0}{\alpha_3 a_1^2 + \beta_1 a_1 + \beta_0}, \quad (\text{D.13})$$

where  $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \beta_0, \beta_1$  are expressions in terms of  $s_1, s_2, s_3, H_1, H_2, H_3$ :

$$\begin{aligned} \alpha_3 &= (H_1 - H_2)^2 + (H_2 - H_3)^2 + (H_3 - H_1)^2 \\ \alpha_2 &= (s_1 + s_2 + s_3)(H_1 H_2 + H_1 H_3 + H_2 H_3) \\ &\quad - \frac{3}{2} [s_1(H_2 - H_3)^2 + s_2(H_3 - H_1)^2 + s_3(H_1 - H_2)^2] \\ \alpha_1 &= s_1^2(H_2 - H_3)^2 + s_2^2(H_1 - H_3)^2 + s_3^2(H_1 - H_2)^2 \\ &\quad - 2s_1(H_2 - H_3)(s_2 H_3 - s_3 H_2) - 2s_2(H_3 - H_1)(s_3 H_1 - s_1 H_3) - 2s_3(H_1 - H_2)(s_1 H_2 - s_2 H_1) \\ \alpha_0 &= s_1 s_2 (s_1 H_2 H_3 + s_2 H_1 H_3 - s_1 H_3^2 - s_2 H_3^2) + s_2 s_3 (s_2 H_1 H_3 + s_3 H_1 H_2 - s_2 H_1^2 - s_3 H_1^2) \\ &\quad + s_1 s_3 (s_1 H_2 H_3 + s_3 H_1 H_2 - s_1 H_2^2 - s_3 H_2^2) \\ \beta_1 &= -s_1(H_2 - H_3)^2 - s_2(H_1 - H_3)^2 - s_3(H_1 - H_2)^2 \\ \beta_0 &= s_1^2(H_2 - H_3)^2 + s_2^2(H_1 - H_3)^2 + s_3^2(H_1 - H_2)^2. \end{aligned} \quad (\text{D.14})$$

Clearly, if  $H_1 \neq H_2 \neq H_3$  (which is implied by our assumption that measurements are distinct), then  $\alpha_3 \neq 0$ . Also, since we assume that measurements are noisy,  $\tilde{c}_1$  will be an order 3 polynomial in the numerator, divided by an order 2 polynomial in the denominator. We have reached convergence when  $\tilde{c}_1$  is 0, so we are essentially looking for the roots of the polynomial in the numerator, namely  $\alpha_3 a_1^3 + \alpha_2 a_1^2 + \alpha_1 a_1 + \alpha_0$ . These will be possible convergence points of the iteration. The VF iteration is (for better layout, we change the notation  $a_1^{(k)}$  to  $a_k$ )

$$a_{k+1} = a_k - \tilde{c}_1 \Rightarrow a_{k+1} = a_k - \underbrace{(\alpha_3 a_k^2 + \beta_1 a_k + \beta_0)^{-1}}_{g(a_k)^{-1}} \cdot \underbrace{(\alpha_3 a_k^3 + \alpha_2 a_k^2 + \alpha_1 a_k + \alpha_0)}_{f(a_k)}. \quad (\text{D.15})$$

Recall that the Newton iteration is of the form

$$a_{k+1} = a_k - f'(a_k)^{-1} f(a_k), \quad (\text{D.16})$$

where  $f'(a_k)$  is the derivative of the polynomial  $f(a_1)$  evaluated at the current iterate  $a_k$ , so for our case, this would be

$$a_{k+1} = a_k - \underbrace{(\alpha_3 a_k^3 + \alpha_2 a_k^2 + \alpha_1 a_k + \alpha_0)}_{f(a_k)} \cdot \underbrace{(3\alpha_3 a_k^2 + 2\alpha_2 a_k + \alpha_1)^{-1}}_{f'(a_k)^{-1}}. \quad (\text{D.17})$$

On the other hand, we can also have a Newton iteration of the form

$$a_{k+1} = a_k - \left( \frac{\partial}{\partial a_1} \left( \frac{f}{g} \right) \Big|_{a_1=a_k} \right)^{-1} \frac{f(a_k)}{g(a_k)} = a_k - \frac{f(a_k)g(a_k)}{f'(a_k)g(a_k) - f(a_k)g'(a_k)}. \quad (\text{D.18})$$

For the Vector Fitting iteration,  $h(a) = a - \frac{f(a)}{g(a)}$ , therefore,

$$h'(a) = 1 - \frac{f'(a)g(a) - f(a)g'(a)}{g^2(a)}, \quad (\text{D.19})$$

which, when evaluated at those  $a$  which make  $f(a) = 0$  leads to

$$h'(a) = 1 - \frac{f'(a)}{g(a)} = 1 - \frac{3\alpha_3 a^2 + 2\alpha_2 a + \alpha_1}{\alpha_3 a^2 + \beta_1 a + \beta_0} = \frac{-2\alpha_3 a^2 + (\beta_1 - 2\alpha_2)a + (\beta_0 - \alpha_1)}{\alpha_3 a^2 + \beta_1 a + \beta_0}. \quad (\text{D.20})$$

Note that in the case of the Newton iteration,  $h(a) = a - \frac{f(a)}{f'(a)}$ , so

$$h'(a) = 1 - \frac{f'(a)f'(a) - f(a)f''(a)}{(f'(a))^2} = -\frac{f(a)f''(a)}{(f'(a))^2}, \quad (\text{D.21})$$

which, when evaluated at those  $a$  which make  $f(a) = 0$  leads to  $h'(a) = 0$ , so all solutions of  $f(a) = 0$  are attractive solutions and, thus, possible convergence points. Therefore, the Newton iteration can converge to any of these solutions, depending on the initial guess for the starting pole.

Fig. D.1 shows the “basins of attraction” associated to the Newton iterations in the numerical examples D.1.1 and D.1.2. Each plot shows the 3 possible solutions (the roots of the polynomial  $f$ ). Also, the regions of the complex plane which, when used as a starting guess for the Newton iteration, converge to the corresponding solution, are color coded. For example, starting from a region where the color code is green leads to the real solution in each plot in Fig. D.1.

**Example D.1.1.** *The underlying transfer function is  $H(s) = \frac{2}{s+5}$  and we take 3 measurements at  $s = 1$ ,  $s = 2$  and  $s = 3$ , to which we add .01 for the odd measurements and  $-.01$  for the second measurement. Therefore, the measurements are  $H_1 = \frac{2}{6} + \frac{1}{100} = \frac{103}{300}$ ,  $H_2 = \frac{2}{7} - \frac{1}{100} = \frac{193}{700}$ ,  $H_3 = \frac{2}{8} + \frac{1}{100} = \frac{13}{50}$ . This leads to the following VF iteration:*

$$a_{k+1} = a_k - \underbrace{\frac{51878a_k^3 - 7293a_k^2 - 740839a_k + 1216226}{51878a_k^2 - 245662a_k + 305065}}_{\tilde{z}_1}, \quad (\text{D.22})$$

so the polynomial we are trying to find the roots of is  $51878a^3 - 7293a^2 - 740839a + 1216226$ . The 3 roots computed in Matlab using the command `roots` are  $-4.36$  and  $2.25 \pm 0.55i$ . On the other hand, the 2 roots of the polynomial in the denominator (namely  $51878a^2 - 245662a + 305065$ ) are  $2.36 \pm 0.52i$ . Recall that in the case of

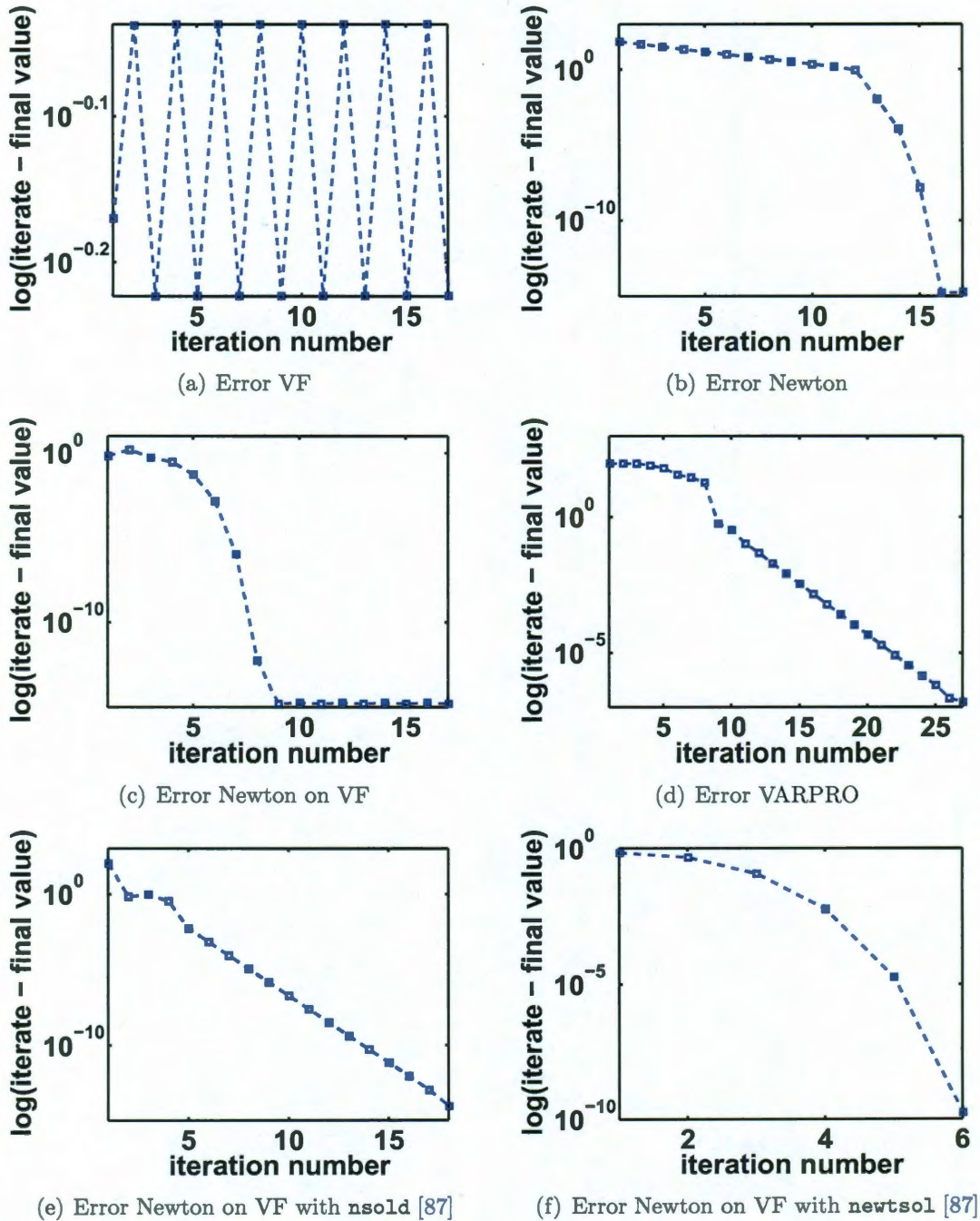
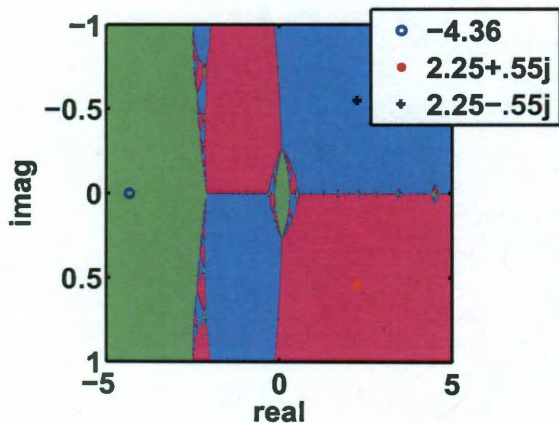
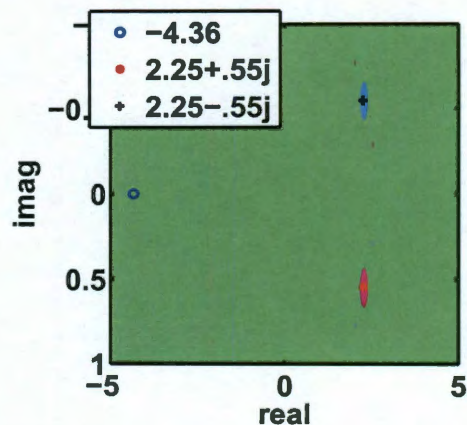


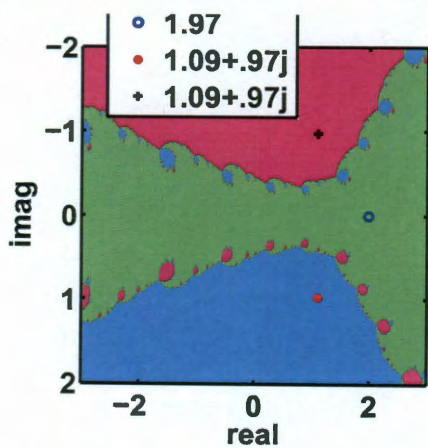
Figure D.3: Errors for the various methods considered



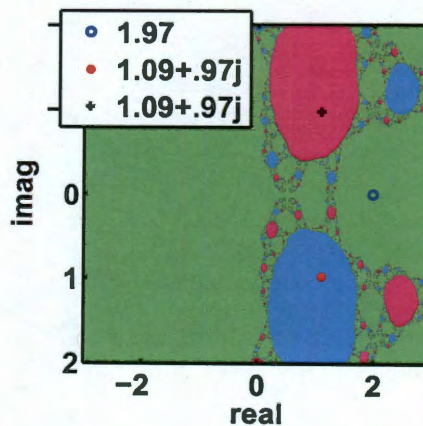
(a) Fractal for Newton iteration in Example D.1.1



(b) Fractal for Newton on VF in Example D.1.1



(c) Fractal for Newton iteration in Example D.1.2



(d) Fractal for Newton on VF in Example D.1.2

Figure D.1: Fractals for the Newton iteration

*Newton iteration converges slower than the Newton step applied to the VF iteration. VARPRO exhibits a linear convergence rate after iteration 10.*

Based on these numerical examples, we are ready to provide an analysis of the conditions which make the vector fitting iteration exhibit a single attractive fixed point. Suppose the fixed points of the VF iteration, namely the roots of  $f(a)$  (see Eq.(D.15)), are  $q$ ,  $p_1 + \Delta p_1$  and  $p_2 + \Delta p_2$ , so  $f(a) = (a - q)[a - (p_1 + \Delta p_1)][a - (p_2 + \Delta p_2)]$ . The root  $q$  can be thought of as the attractive root, while  $p_1 + \Delta p_1$  and  $p_2 + \Delta p_2$  as the repelling ones. Recall that in Example D.1.1 the roots of the denominator  $g(a)$  were close to the two repelling fixed points. Thus, we choose to express  $g(a)$  as  $(a - p_1)(a - p_2)$ . Clearly, if there is no noise in the data, then  $p_1 + \Delta p_1 = p_2 + \Delta p_2 = 0$  and  $\tilde{c}_1 = \frac{f(a)}{g(a)} = a - q$  leading to  $h(a) = a - \tilde{c}_1 = a - (a - q) = q$  which turns out to be the pole of the underlying system. When  $p_1 + \Delta p_1 \neq 0$  and  $p_2 + \Delta p_2 \neq 0$ , we compute  $h'(a)$  and evaluate it at each root of  $f(a)$  (namely,  $q$ ,  $p_1 + \Delta p_1$  or  $p_2 + \Delta p_2$ ), but for now we still use the notation  $a$ ). Substituting into Eq.(D.20) we obtain

$$\begin{aligned} h'(a) &= 1 - \frac{f'(a)}{g(a)} \\ &= 1 - \frac{[a - (p_1 + \Delta p_1)][a - (p_2 + \Delta p_2)] + (a - q)[a - (p_1 + \Delta p_1)] + (a - q)[a - (p_2 + \Delta p_2)]}{(a - p_1)(a - p_2)}. \end{aligned}$$

We are now ready to evaluate  $h'(q)$ ,  $h'(p_1 + \Delta p_1)$  and  $h'(p_2 + \Delta p_2)$ .

$$\begin{aligned} h'(q) &= 1 - \frac{[q - (p_1 + \Delta p_1)][q - (p_2 + \Delta p_2)]}{(q - p_1)(q - p_2)} \\ &= 1 - \left(1 - \frac{\Delta p_1}{q - p_1}\right) \left(1 - \frac{\Delta p_2}{q - p_2}\right). \end{aligned}$$

Recall that  $q$  was the notation for the attractive solution, so we would like for  $|h'(q)|$  to be less than 1. This can be achieved if  $\frac{\Delta p_1}{q - p_1}$  and  $\frac{\Delta p_2}{q - p_2}$  have magnitude smaller than  $\sqrt{2} - 1$ . Therefore, if the noise in the data is small, then the roots of  $f(a)$  will be very close to those of  $g(a)$  and, consequently,  $\Delta p_1$  and  $\Delta p_2$  will be several orders of



noise-free measurements, the pole at the first iteration is given by

$$a_1 = a_1 - \underbrace{(a_1 - \beta)}_{\tilde{c}_1} = a_1 - \underbrace{\frac{a_1 - \beta}{1}}_{\tilde{c}_1}, \quad (\text{D.23})$$

so the polynomial in the denominator of  $\tilde{c}_1$  is 1 (which is an order 0 polynomial, as opposed to order 2 for noisy measurements as in Eq.(D.13)). For noisy measurements, we expect that the roots of the polynomial in the denominator are close to those of the polynomial in the numerator. The difference in these quantities is  $-0.11 \pm 0.03i$ .

Clearly, the solution we would like VF to converge to is  $-4.36$ , since this will be the recovered pole at the end of the iteration (recall that the pole of the underlying system is  $-5$ , but due to the noise we are adding, we expect to recover something in the neighborhood of  $-5$ ).

We can check whether each of the above solutions are attractive, repelling, or indifferent fixed points based on the criterion  $|h'(a)| < 1$ ,  $|h'(a)| > 1$  or  $|h'(a)| = 1$ , respectively, where  $h(a) = a - \frac{51878a^3 - 7293a^2 - 740839a + 1216226}{51878a^2 - 245662a + 305065}$ . In our case, the values are  $0.03$ ,  $57.1 \pm 10.6i$ , so, as expected, the fixed point  $-4.36$  is attractive, while the other two, namely  $2.25 \pm 0.55i$ , are repelling. Therefore, irrespective of the initial guess we use for the starting pole, the VF iteration will converge to the attractive solution, unless the initial starting pole, namely the first guess, coincides with one of the repelling solutions, in which case the iterates will not change.

Interestingly, Eq.(D.22) can be rewritten as

$$a_{k+1} = \frac{-238369a_k^2 + 1045904a_k - 1216226}{51878a_k^2 - 245662a_k + 305065} \quad (\text{D.24})$$

and, when dividing the coefficients of the same powers of  $a_k$ , we obtain  $-\frac{238369}{51878} = -4.59$ ,  $-\frac{1045904}{245662} = -4.26$  and  $-\frac{1216226}{305065} = -3.99$ , which are all in the neighborhood of the attractive fixed point located at  $-4.36$ .

magnitude smaller than  $q - p_1$  and  $q - p_2$ . On the other hand,

$$\begin{aligned} h'(p_1 + \Delta p_1) &= 1 - \frac{[(p_1 + \Delta p_1) - q] [(p_1 + \Delta p_1) - (p_2 + \Delta p_2)]}{\Delta p_1 (p_1 + \Delta p_1 - p_2)} \\ &= 1 + \left( \frac{\Delta p_2}{p_1 - p_2 + \Delta p_1} - 1 \right) \left( \frac{p_1 - q}{\Delta p_1} + 1 \right). \end{aligned}$$

Recall that  $p_1 + \Delta p_1$  was the notation for one of the repelling solutions, so we would like for  $|h'(p_1 + \Delta p_1)|$  to be larger than 1. This can be achieved if  $\frac{p_1 - q}{\Delta p_1}$  has magnitude larger than  $\sqrt{2} + 1$  and  $\frac{\Delta p_2}{p_1 - p_2 + \Delta p_1}$  has magnitude smaller than  $\sqrt{2} - 1$ , which is consistent with the condition obtained from  $h'(q) < 1$ .

Therefore, if the noise in the data is small, then the roots of the numerator  $f(a)$  will be very close to those of the denominator  $g(a)$  and, consequently,  $\Delta p_1$  and  $\Delta p_2$  will be several orders of magnitude smaller than  $p_1 - q$  and  $p_1 - p_2$ . One can derive a similar expression for  $h'(p_2 + \Delta p_2)$  since the expression for  $h'(a)$  is symmetric in  $p_1 + \Delta p_1$  and  $p_2 + \Delta p_2$ .

Next, we derive the solution one obtains when solving the least squares system set up using the *Lagrange basis* (see Eq.(6.66)). This time,  $\tilde{c}_1 = a_1 - \frac{\alpha_0}{\beta_0}$ , where  $\alpha_0, \beta_0$  are expressions in terms of  $s_1, s_2, s_3, H_1, H_2, H_3$ . Since we want  $\tilde{c}_1$  to be 0, we are looking for the solution of  $a_1 - \frac{\alpha_0}{\beta_0} = 0$ , which is  $a_1 = \frac{\alpha_0}{\beta_0}$ . Therefore, in this case, there is no iteration, and irrespective of the starting pole chosen as an initial guess, the recovered pole is found as

$$a_1 = \frac{(H_2 + H_3 - 2H_1)H_1s_1 + (H_1 + H_3 - 2H_2)H_2s_2 + (H_1 + H_2 - 2H_3)H_3s_3}{-2H_1^2 - 2H_2^2 - 2H_3^2 + 2H_1H_2 + 2H_1H_3 + 2H_2H_3}, \quad (\text{D.28})$$

which is determined entirely from the available measurements and does not depend on the initial starting pole.

**Example D.1.3.** *We consider the same example as before. The value of the pole obtained using the Lagrange basis (using Eq.(D.28)) is  $-4.59$ . This is closer to the*

Table D.1 summarizes the iterates at each step using a starting pole located at  $-100$  as initial guess and the following methods: vector fitting, the Newton iteration considering the derivative of the polynomial in the numerator  $f$ , the Newton iteration considering the derivative of the fraction  $\frac{f}{g}$  (referred to as “Newton on VF” in the table) and, lastly, VARPRO, with an implementation provided by Jeff Hokanson [86].

Iteration	VF	Newton	Newton on VF	VARPRO
1	-4.5792	-66.6836	-4.5647	-99.9998
2	-4.3704	-44.4894	-4.3636	-99.9976
3	-4.3637	-29.7193	-4.3635	-99.9756
4	-4.3635	-19.9131	-4.3635	-99.7559
5	-4.3635	-13.4400	-4.3635	-97.5526
6	-4.3635	-9.2290	-4.3635	-74.9322
7	-4.3635	-6.5928	-4.3635	-38.9679
8	-4.3635	-5.1087	-4.3635	-27.4569
9	-4.3635	-4.4884	-4.3635	-9.6986
10	-4.3635	-4.3679	-4.3635	-5.7283
11	-4.3635	-4.3635	-4.3635	-4.8537
12	-4.3635	-4.3635	-4.3635	-4.7750
13	-4.3635	-4.3635	-4.3635	-4.7717
14	-4.3635	-4.3635	-4.3635	-4.7716
15	-4.3635	-4.3635	-4.3635	-4.7716

Table D.1: Poles at each iteration

The first three columns are all solving the same problem, but in different ways. The vector fitting iteration converges to the solution in 4 iterations, as opposed to the Newton iteration, which converges in 11 iterations and the Newton step applied to the vector fitting iteration, which converges in 3 steps. The last column shows the results obtained with VARPRO, the method explained in Sect.6.2. This converges in 14 steps (because the step size is adjusted at each iteration) and the recovered pole is closer to the true pole (located at  $-5$ ).

Fig. D.2 presents the absolute error at each step, between the current iterate and the final iterate, on a logarithmic scale, for all 4 methods considered. Vector fitting exhibits linear convergence. The traditional Newton iteration converges very slow

original pole located at  $-5$  than  $-4.36$ , the value found at the end of the VF iteration.

## D.2 Order 2 strictly proper system

Next, we discuss the case of an order 2 strictly proper system. The underlying system is of the form

$$H(s) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2}. \quad (\text{D.29})$$

We use two starting poles for the model,  $a_1, a_2$ , so the expression we obtain is

$$f(s) = \frac{\frac{c_1}{s-a_1} + \frac{c_2}{s-a_2}}{\frac{\tilde{c}_1}{s-a_1} + \frac{\tilde{c}_2}{s-a_2} + 1} = \frac{s(c_1 + c_2) - c_1a_2 - c_2a_1}{s^2 + s(\tilde{c}_1 + \tilde{c}_2 - a_1 - a_2) + (a_1a_2 - \tilde{c}_1a_2 - \tilde{c}_2a_1)}. \quad (\text{D.30})$$

The poles at the new VF iteration are given by the 2 solutions of the polynomial equation  $s^2 + s(\tilde{c}_1 + \tilde{c}_2 - a_1 - a_2) + (a_1a_2 - \tilde{c}_1a_2 - \tilde{c}_2a_1) = 0$ . Changing the notation to introduce the iteration count, the poles at iteration  $k + 1$ , namely  $a_1^{(k+1)}$  and  $a_2$ , are given as

$$a_{1,2}^{(k+1)} = \frac{-(\tilde{c}_1 + \tilde{c}_2 - a_1^{(k)} - a_2^{(k)}) \pm \sqrt{(\tilde{c}_1 + \tilde{c}_2 - a_1^{(k)} - a_2^{(k)})^2 - 4(a_1^{(k)}a_2^{(k)} - \tilde{c}_1a_2^{(k)} - \tilde{c}_2a_1^{(k)})}}{2}, \quad (\text{D.31})$$

where  $\tilde{c}_1, \tilde{c}_2$  are found by solving the least squares problem in Eq.(6.4), which is set up using the measurements, as well as the  $k^{\text{th}}$  iterates  $a_1^{(k)}$  and  $a_2^{(k)}$ .

Eq.(D.29) contains 4 unknowns, namely  $r_1, r_2, p_1, p_2$ , so a minimum of 4 conditions are required to completely determine them. Therefore, at least 4 samples and their corresponding measurements are needed. In case we only have 4 measurements, Eq.(6.4) lead to the solution  $\mathbf{A}^{-1}\mathbf{b}$ , since the matrix  $\mathbf{A}$  is square.

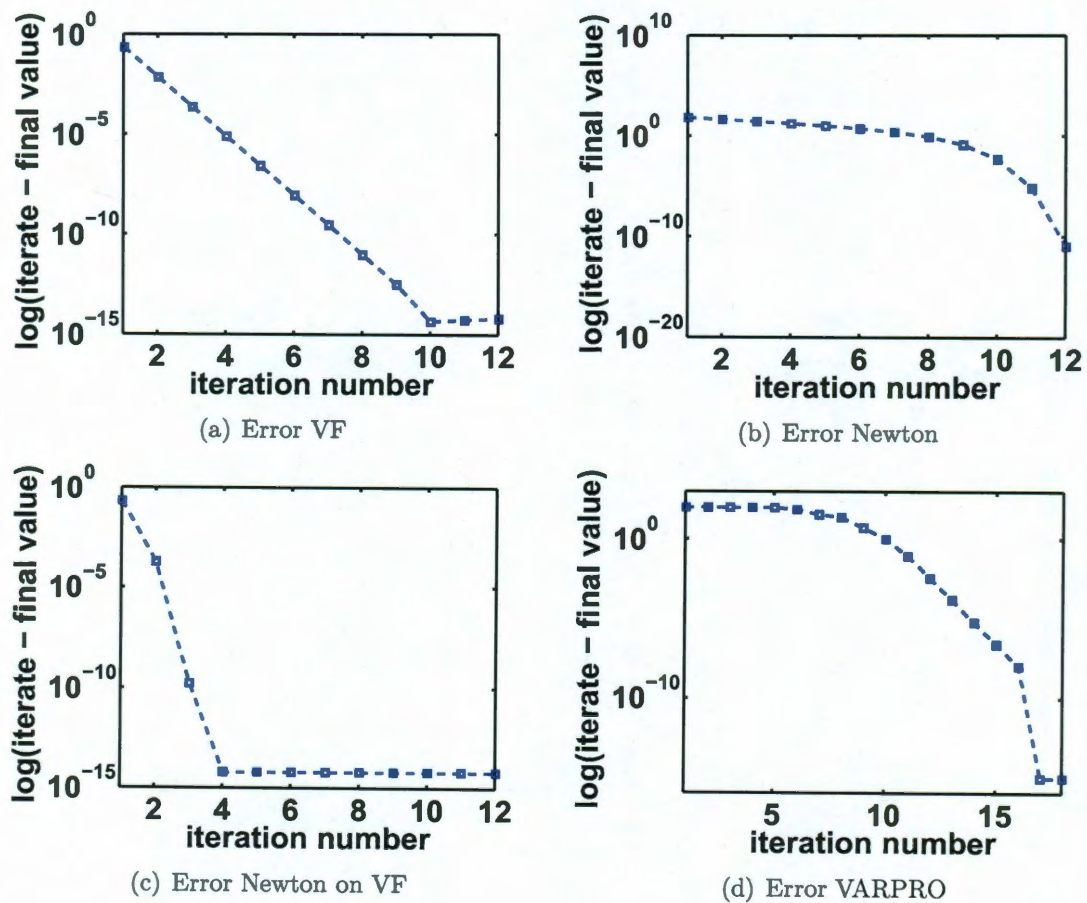


Figure D.2: Errors for the various methods considered

### D.2.1 5 measurements

Suppose we have 5 measurements available of  $H(s)$  at  $s_1, s_2, s_3, s_4$  and  $s_5$ , namely  $H_1, H_2, H_3, H_4$  and  $H_5$ . The linear system we are solving is  $\mathbf{A}(s_i, 5; a_i, 2)\mathbf{x}(c_i, 2; \tilde{c}_i, 2) = \mathbf{b}$ . The system is overdetermined, so the solution  $\mathbf{x}$  is found via least squares. Next, the starting poles at the next iteration are given by Eq.(D.31). When convergence is reached,  $\tilde{c}_1 = \tilde{c}_2 = 0$ . Using the fact that  $\tilde{c}_1, \tilde{c}_2$  are rational expressions in  $a_1, a_2$ , we are looking for all pairs of solutions  $(a_1^{(s)}, a_2^{(s)})$  which make  $\tilde{c}_1 = \tilde{c}_2 = 0$ .

Using the normal equations, the least squares solution is given as  $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ .

It turns out that

$$\tilde{c}_1 = \frac{a_1^4(\gamma_1 a_2^2 + \gamma_2 a_2 + \gamma_3) + a_1^3(\gamma_4 a_2^2 + \gamma_5 a_2 + \gamma_6) + \dots}{a_1^3(\delta_1 a_2^2 + \delta_2 a_2 + \delta_3) + a_1^2(\delta_4 a_2^3 + \delta_5 a_2 + \delta_6) + \dots} = \frac{f_1(a_1, a_2)}{g(a_1, a_2)}, \quad (\text{D.32})$$

$$\tilde{c}_2 = -\frac{a_2^4(\gamma_1 a_1^2 + \gamma_2 a_1 + \gamma_3) + a_2^3(\gamma_4 a_1^2 + \gamma_5 a_1 + \gamma_6) + \dots}{a_1^3(\delta_1 a_2^2 + \delta_2 a_2 + \delta_3) + a_1^2(\delta_4 a_2^3 + \delta_5 a_2 + \delta_6) + \dots} = \frac{f_2(a_1, a_2)}{g(a_1, a_2)}, \quad (\text{D.33})$$

where ... stand for the rest of the terms in the expressions, which contain powers of  $a_1$  and  $a_2$  lower than the ones listed above and  $\delta_i, \gamma_i$  are complicated expressions in terms of  $s_1, s_2, s_3, s_4, s_5$ , and the corresponding measurements  $H_1, H_2, H_3, H_4, H_5$ . We want  $\tilde{c}_1$  and  $\tilde{c}_2$  to be 0 when convergence is reached, so we are looking for the roots of the coupled multivariate polynomials in the numerator:  $f_1(a_1, a_2) = 0$  and  $f_2(a_1, a_2) = 0$ . One can use a Newton step on the vector fitting iteration to improve the convergence properties of vector fitting in terms of the number of iterations needed to reach convergence, but also, to ensure that the iteration will always converge. In this case, the iteration is of the form

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \mathbf{J}_{f/g}(\mathbf{a}_k)^{-1} \left( \frac{1}{g(\mathbf{a}_k)} \mathbf{f}(\mathbf{a}_k) \right), \quad (\text{D.34})$$

with  $\mathbf{J}_{f/g}(\mathbf{a}_k)$ , the Jacobian matrix, defined as  $\mathbf{J}_{f/g, i, j} = \frac{\partial(g^{-1} \mathbf{f}_i)}{\partial(\mathbf{a}_j)}$ .

**Example D.2.1.** *Let us analyze the following numerical example. The underlying*

because the initial guess is far away from the solution, but once in the neighborhood of the convergence point, it converges quadratically. The Newton step applied to the VF iteration also shows quadratic convergence, even though the initial guess is poor. Lastly, VARPRO shows linear convergence between iterations 10 and 15.

Recall that the original Newton iteration converges to any of the 3 possible solutions (computed as the roots of the polynomial in the numerator), depending on the initial guess of the starting pole. Clearly, since the other two possible convergence points are complex, the iteration will converge to the real solution for any real starting pole. Table D.2 lists the iterates when a starting pole located at  $10j$  is used as initial guess.

Iteration	Iterate VF	Iterate Newton	Iterate Newton on VF	Iterate VARPRO
1	$-4.55 + 0.15i$	$0.08 + 6.36i$	$-4.42 + 0.24i$	$-0.02 + 10.007i$
2	$-4.36 + 4e-3i$	$0.24 + 3.79i$	$-4.36 + 1e-4i$	$-0.26 + 10.07i$
3	$-4.36 + 1e-4i$	$0.58 + 1.93i$	$-4.36 - 4e-10i$	$-2.37 + 10.59i$
4	$-4.36 + 5e-6i$	$1.25 + .75i$	$-4.36 - 9e-22i$	$-11.36 + 9.43i$
5	$-4.36 + 1e-7i$	$1.88 + .37i$	$-4.36 + 3e-37i$	$-11.54 + 4.15i$
6	$-4.36 + 5e-9i$	$2.29 + .39i$	$-4.36 - 1e-52i$	$-8.41 + 0.43i$
7	$-4.36 + 1e-10i$	$2.22 + .57i$	$-4.36 + 7e-68i$	$-3.39 - 0.41i$
8	$-4.36 + 6e-12i$	$2.25 + .54i$	$-4.36 - 3e-83i$	$-4.47 - 0.14i$
9	$-4.36 + 2e-13i$	$2.25 + .54i$	$-4.36 + 1e-98i$	$-4.74 - 6e-3i$
10	$-4.36 + 7e-15i$	$2.25 + .54i$	$-4.36 - 6e-114i$	$-4.77 + 2e-4i$
11	$-4.36 + 2e-16i$	$2.25 + .54i$	$-4.36 + 2e-129i$	$-4.77 - 9e-6i$
12	$-4.36 + 7e-18i$	$2.25 + .54i$	$-4.36 - 1e-144i$	$-4.77 + 4e-7i$

Table D.2: Poles at each iteration

The vector fitting iteration converges to the desired solution in 10 iterations, as opposed to the original Newton iteration, which converges one of the other two fixed points. However, the Newton step applied to the vector fitting iteration converges in 4 steps to the desired solution, located at  $-4.36$ . This table supports the fractals in Fig. D.1(a) - D.1(a), namely that, when applying the Newton iteration to the vector fitting expression, which only has one attractive point located at  $-4.36$ , the areas in the complex plane which converge to one of the other two possible solutions (which are repelling for VF) shrink.

transfer function is

$$\dot{H}(s) = \frac{2}{s+5} + \frac{3}{s+10}.$$

We take 5 measurements at  $s = 1$ ,  $s = 2$ ,  $s = 3$ ,  $s = 4$  and  $s = 5$ , and corrupt them with additive noise generated with the following line of code in Matlab:

`H.*(10-4)*randn(size(H))`,

so the added noise is  $10^{-4}$  relative to the magnitude of each entry, yielding an iteration as in Eq.(D.31). At convergence we have that  $\tilde{c}_1 = \tilde{c}_2 = 0$ , so we aim at finding the roots of the coupled multivariate polynomials in the numerator, namely

$$\begin{aligned} & a_1^4 a_2^2 - 7.65 a_1^4 a_2 + 15.34 a_1^4 + 5.82 a_1^3 a_2^2 - 47.39 a_1^3 a_2 + 91.68 a_1^3 - 53.24 a_1^2 a_2^2 + 403.1 a_1^2 a_2 \\ & - 809.34 a_1^2 - 125.4 a_1 a_2^2 + 999.21 a_1 a_2 - 1872.2 a_1 + 691.92 a_2^2 - 5501.02 a_2 + 10987.04 = 0 \end{aligned} \quad (\text{D.35})$$

$$\begin{aligned} & a_1^2 a_2^4 + 5.82 a_1^2 a_2^3 - 53.24 a_1^2 a_2^2 - 125.4 a_1^2 a_2 + 691.92 a_1^2 - 7.65 a_1 a_2^4 - 47.39 a_1 a_2^3 + 403.11 a_1 a_2^2 \\ & + 999.21 a_1 a_2 - 5501.02 a_1 + 15.34 a_2^4 + 91.68 a_2^3 - 809.34 a_2^2 - 1872.22 a_2 + 10987.04 = 0. \end{aligned} \quad (\text{D.36})$$

The two expressions are symmetric in the variables  $a_1$  and  $a_2$ . The roots are listed in Table D.4, together with the spectral radius of the Jacobian matrix evaluated at each solution. Clearly, the first two pairs are attractive solutions, since the norm of the Jacobian is less than 1, while the rest are repelling solutions since the norm of the Jacobian is larger than 1. The results in the last 5 rows make sense since having the two starting poles equal to each other leads to a Jacobian matrix with 2 linearly dependent columns. We conclude that the poles of the model found at the end of the VF iteration are going to be  $-4.69$  and  $-9.36$  (note that their order is not important). Therefore, irrespective of the initial guess we use for the starting poles, we converge to the attractive solution, unless we start the iteration with one of the repelling solutions as our guess, in which case the iterates will not change. Also, note that the attractive



**Example D.1.2.** We assume a system with an underlying transfer function given by  $H(s) = \frac{2}{s+5}$  and consider 3 measurements at  $s = 1$ ,  $s = 2$  and  $s = 3$ , to which we add noise generated using the following line of code in Matlab

```
H+abs(H).*rand(1,Ns)*10^-1,
```

that is, we add random noise normally distributed with zero mean and standard deviation 10 relative to the magnitude of each measurement. The measurements are  $H_1 = 1.64$ ,  $H_2 = 2.16$ ,  $H_3 = .68$ . This leads to the following VF iteration:

$$a_{k+1} = a_k - \frac{a_k^3 - 4.16a_k^2 + 6.46a_k - 4.23}{a_k^2 - 2.86a_k + 2.45} = a_k - \frac{f(a_k)}{g(a_k)}, \quad (\text{D.25})$$

so the polynomial we are trying to find the roots of is  $a^3 - 4.16a^2 + 6.46a - 4.23$ . The 3 roots computed in Matlab using the command `roots` are 1.97 and  $1.09 \pm .97i$ . On the other hand, the 2 roots of the polynomial in the denominator (namely  $a^2 - 2.86a + 2.45$ ) are  $1.43 \pm .63i$  and the difference between these and the roots of the numerator is  $-.34 \pm .34i$  (with respect to  $1.09 \pm .97i$ ) and  $.54 \pm .63i$  (with respect to 1.97).

When computing  $h'(a)$ , where  $h(a) = a - \frac{a^3 - 4.16a^2 + 6.46a - 4.23}{a^2 - 2.86a + 2.45}$ , to check which of the fixed points are attractive, repelling, or indifferent, we obtain:  $h'(1.97) = -1.48$ ,  $h'(1.09 \pm .97i) = -2.14 \pm .83i$ , therefore, in this case, all roots are repelling, so irrespective of the initial starting pole, the vector fitting iteration will not converge.

The original Newton iteration in this case is

$$a_{k+1} = a_k - \frac{a_k^3 - 4.16a_k^2 + 6.46a_k - 4.23}{3a_k^2 - 8.32a_k + 6.46} = a_k - \frac{f(a_k)}{f'(a_k)}, \quad (\text{D.26})$$

while the one applied to the VF iteration is

$$a_{k+1} = a_k - \left( \frac{\partial}{\partial a_1} \left( \frac{f}{g} \right) \Big|_{a_1=a_k} \right)^{-1} \frac{f(a_k)}{g(a_k)} = a_k - \frac{f(a_k)g(a_k)}{f'(a_k)g(a_k) - f(a_k)g'(a_k)}. \quad (\text{D.27})$$

Table D.3 lists the iterates using the same initial guess located at  $-100$ .

As expected from our previous observation that all solutions are repelling, the

roots we found are relatively close to the poles of the underlying system, which are located at  $-5$  and  $-10$ .

$a_1$	$a_2$	$\ J\ $
-9.36	-4.69	7.4e-3
-4.69	-9.36	7.4e-3
4.28	-6.75	7.5
-6.75	4.28	7.5
3.76	-6.74	9.99
-6.74	3.76	9.99
$4.1 \pm .49i$	$.77 \mp 1.01i$	$1.04e+4$
$.77 \pm 1.01i$	$4.1 \mp .49i$	$1.04e+4$
$4.22 \pm .58i$	$2.12 \pm .28i$	$2.15e+4$
$2.12 \pm .28i$	$4.22 \pm .58i$	$2.15e+4$
5.96	5.96	$7.36e+14$
3.28	3.28	$7.96e+14$
$3.31 \pm 1.34i$	$3.31 \pm 1.34i$	$1.06e+15$
-4.71	-4.71	$1.64e+16$
-9.33	-9.33	$1.7e+16$

Table D.4: Convergence points and corresponding spectral radius of the Jacobian evaluated at those points

Fig. D.4(a) and D.4(b) show the errors at each iteration for the vector fitting algorithm. The poles at the end of the iteration process are  $-9.36$  and  $-4.69$ . Fig. D.4(c) and D.4(c) show the errors at each iteration for the Newton iteration applied to the vector fitting expression. This approach finds the same solution as VF. Last, Fig. D.4(e) and D.4(f) show the iterates at each step when applying VARPRO to the same data set. Note that the solution found is  $2.57$  and  $-6.21$ , with an unstable pole. Generally, VARPRO yields poles which are closer to the location of the underlying ones. The reason for the poor performance for this example is the low number of measurements considered (only 5), as adding the same relative noise to more measurements leads to poles closer to the expected ones. All methods used an initial set of poles located at  $-15$  and  $-20$ , respectively.

Appendix D.3 discusses scalar numerical examples in which the performance of the

Iteration	Iterate VF	Iterate Newton	Iterate Newton on VF	VARPRO
1	1.29	-66.2	1.30	-99.9
2	2.89	-43.6	3.55	-99.8
3	1.37	-28.6	1.42	-98.5
4	2.89	-18.6	2.27	-84.9
5	1.37	-11.9	1.92	-67.1
6	2.89	-7.49	1.97	-39.4
7	1.37	-4.51	1.97	-32.01
8	2.89	-2.51	1.97	-21.5
9	1.37	-1.16	1.97	-3.19
10	2.89	-.22	1.97	-2.29
11	1.37	.47	1.97	-2.73
12	2.89	1.09	1.97	-2.58
13	1.37	1.98	1.97	-2.65
14	2.89	1.97	1.97	-2.62

Table D.3: Poles at each iteration

*vector fitting iteration does not converge (in fact, it oscillates between two values). The original Newton iteration converges in 15 steps, while the Newton step applied to the VF iteration converges in 7 steps. Note that the recovered pole in this case is unstable. On the other hand, VARPRO converges to a value which is located in the left half plane and is closer to the pole of the underlying system (located at  $-5$ ).*

*Fig. D.3(a)-D.3(d) present the residuals at each step, computed as the error between the current iterate and the final iterate, on a logarithmic scale, for all 4 methods considered. Fig. D.3(e) and D.3(f) present the residuals when considering different step sizes in the Newton step applied to the VF iteration. Note that the first few iterates of the Newton step applied to VF show some oscillation (see Table D.3), so by not considering the full Newton step in the first few iterations (see Sect.6.6 for details on the length of the Newton step), Fig. D.3(f) shows no oscillation. On the other hand, in Fig. D.3(e), there is still an oscillation, while the iterates converge slower to the solution than in Fig. D.3(c), where the step length was always 1. This shows that `nsold.m` only shrinks the step size when the iterate is close to the solution. Note that the vector fitting iteration is the only one which does not converge. The traditional*

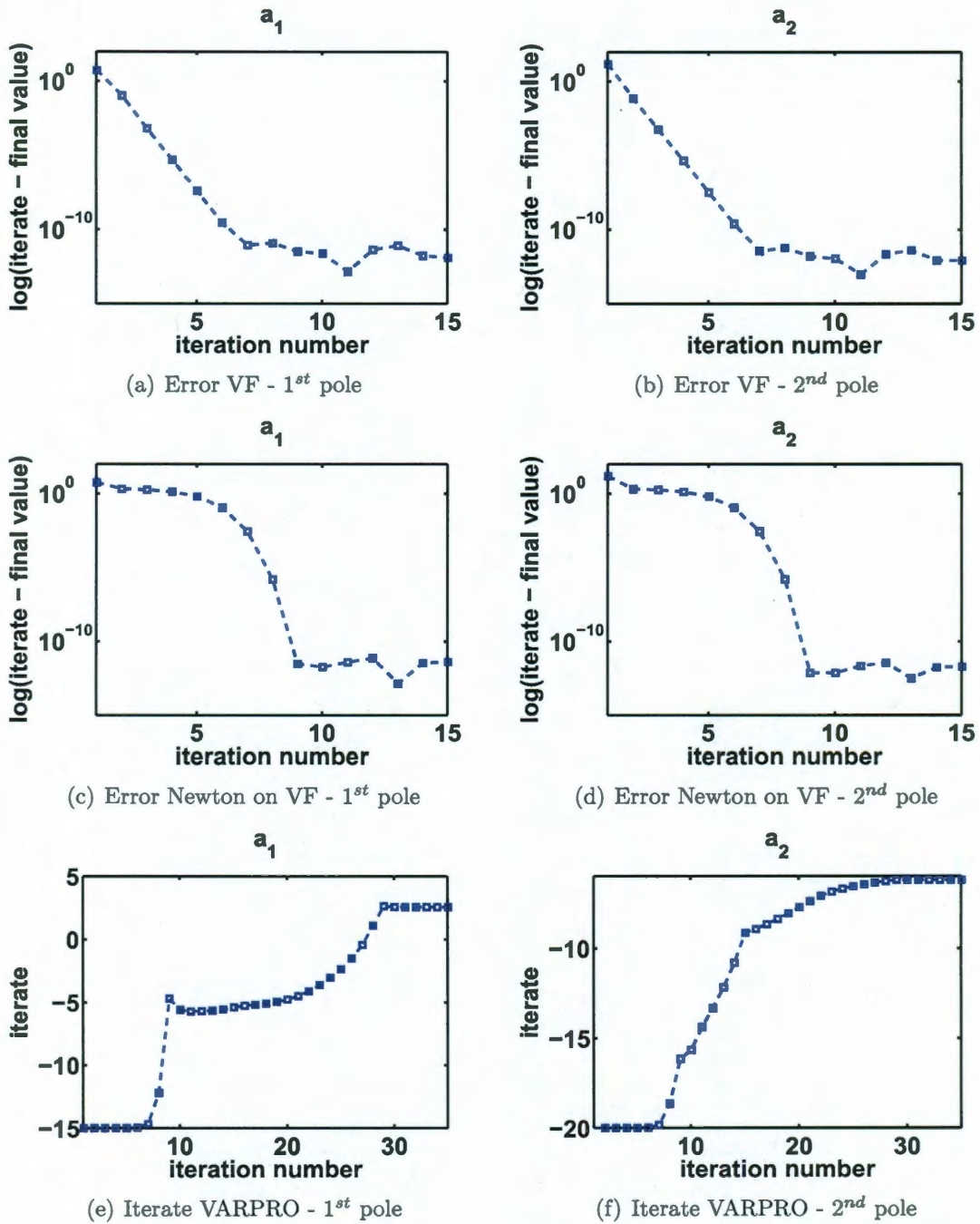


Figure D.4: Errors or iterates for the various methods considered

three methods: the Loewner framework, vector fitting and VARPRO, is compared.

### D.3 Numerical example from S-parameter measurements

We discuss a scalar example obtained from real measurements and compare the three algorithms (Loewner approach in the real alternative implementation (Sect.4.2.3), vector fitting and VARPRO) in terms of the modeling errors and recovered poles. We consider the measurements of the  $S_{1,1}$ -entry of the 50-port example discussed in Sect.4.4.2. Originally, we are provided with 100 measurements, to which we append the measurements at  $-j\omega_i$  as the complex conjugate values of the original ones. Using the real alternative implementation (Sect.4.2.3) of the Loewner framework, we obtain a Loewner matrix pencil for which the decay is shown in Fig. D.5. The Loewner matrix shows a drop after the 4<sup>th</sup> singular value, while the shifted Loewner matrix shows a drop after the 5<sup>th</sup>, indicating that the underlying system is of order 4 with a non-zero  $\mathbf{D}$ -term. The Loewner framework is the only one able to determine the underlying model order out of the three algorithms chosen for comparison.

Truncating to order 5 using the SVD of the Loewner matrix pencil leads to a model which yields the fit shown in Fig. D.7(a). We also compute models of order 5 (with  $\mathbf{D} = \mathbf{0}$ ) with vector fitting (shown in Fig. D.7(b)) and VARPRO (Fig. D.7(c)). All models have a real pole which is far in the left-half plane (Fig. D.6(a)), so we provide a zoom in Fig. D.6(b).

While the model obtained with VARPRO is close to the model obtained with the Loewner framework both in terms of errors (Table D.5), as well as in terms of poles (Fig. D.6), the one obtained with vector fitting has 3 real poles (as opposed to only 1 with the other methods) and the errors are also larger.

Next, we extract the  $\mathbf{D}$ -term from the order 5 model obtained with the Loewner

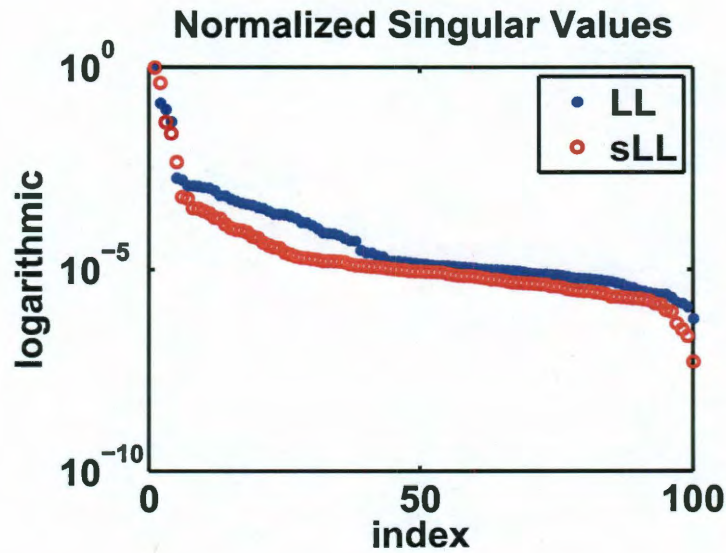


Figure D.5: Singular value drop of the Loewner and shifted Loewner matrices; the Loewner matrix shows a drop after the 4<sup>th</sup> singular value, while the shifted Loewner matrix shows a drop after the 5<sup>th</sup>

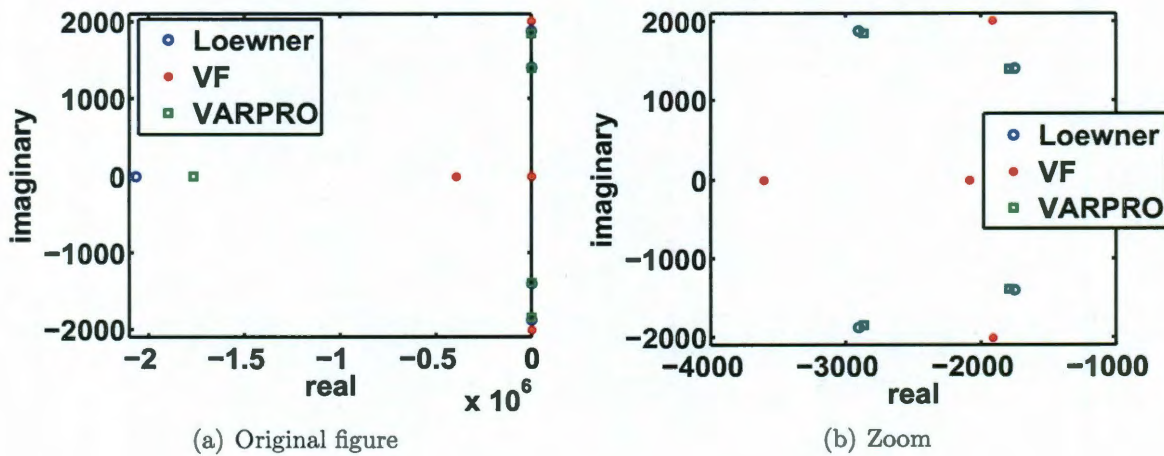
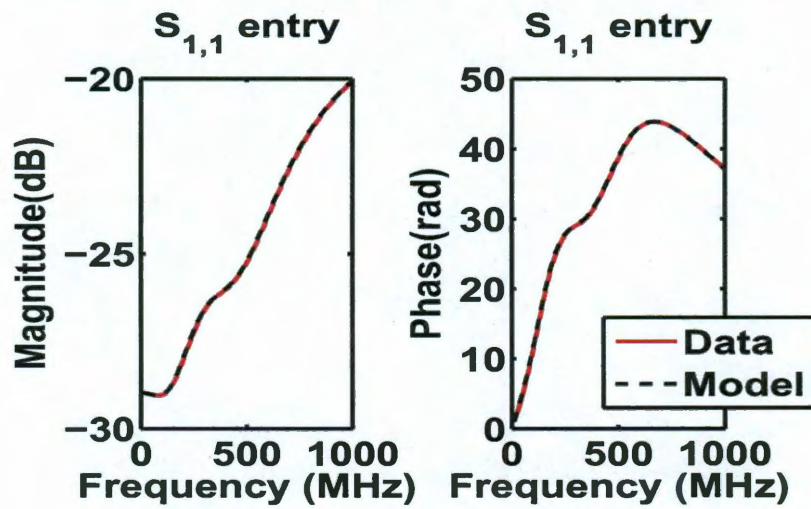


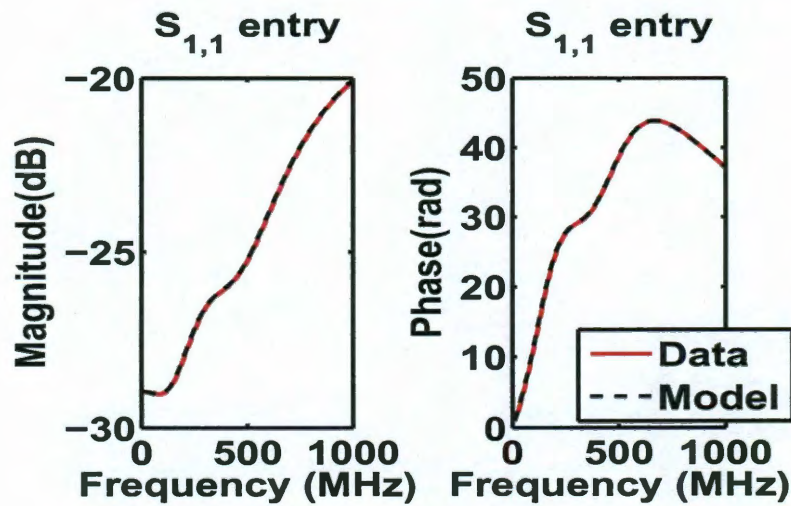
Figure D.6: Poles for models of order 5 with  $D = 0$

Algorithm	$\mathcal{H}_2$ error	$\mathcal{H}_\infty$ error
Loewner	4.39e-4	6.83e-4
Vector Fitting	8.98e-4	1.54e-3
VARPRO	4.26e-4	6.89e-4

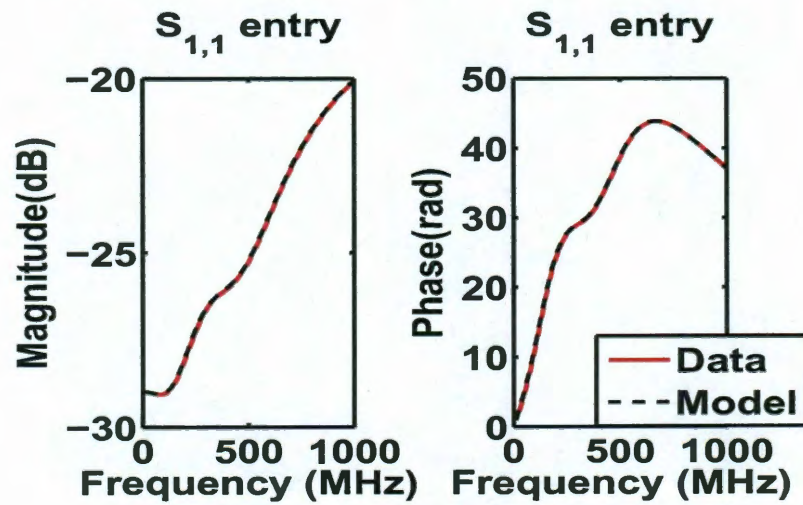
Table D.5: Errors for models of order 5 with  $D = 0$



(a) Loewner



(b) VF



(c) VARPRO

Figure D.7: Model vs. data for order 5 systems with  $D = 0$

framework and compare this order 4 model to the ones obtained with vector fitting and VARPRO, which are now computing models of order 4 with a non-zero  $\mathbf{D}$ -term. The errors are shown in Table D.6 and the poles are shown in Fig. D.8. The poles recovered with vector fitting and VARPRO are at the same location (Fig. D.8), which is also indicated by the similar errors obtained with the two methods. However, these errors are larger than the ones obtained with the Loewner framework and VARPRO for order 5 models with  $\mathbf{D} = \mathbf{0}$ . Therefore, it is preferable for this data set to construct models of order 5 with  $\mathbf{D} = \mathbf{0}$  than models of order 4 with  $\mathbf{D} \neq \mathbf{0}$ .

Algorithm	$\mathcal{H}_2$ error	$\mathcal{H}_\infty$ error
Loewner	2.19e-3	2.64e-3
Vector Fitting	4.85e-4	8.5e-4
VARPRO	4.85e-4	8.44e-4

Table D.6: Errors for models of order 4 with  $\mathbf{D} \neq \mathbf{0}$

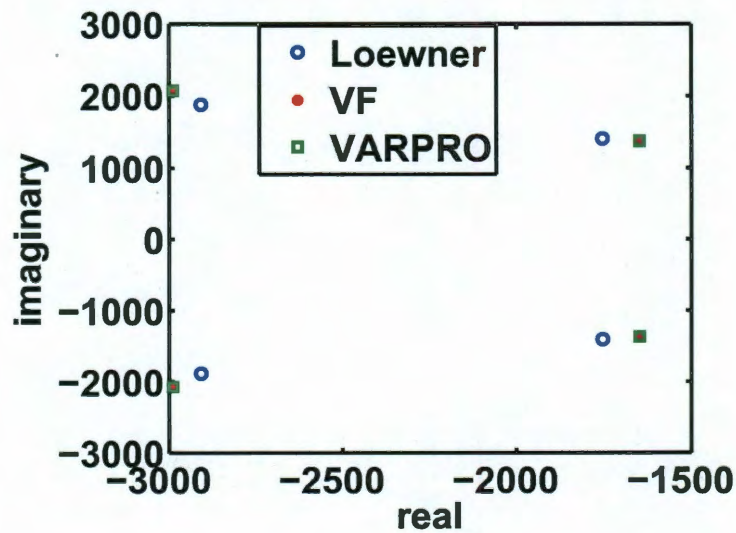


Figure D.8: Poles for models of order 4 with  $\mathbf{D} \neq \mathbf{0}$

We also investigate over-fitting, by constructing models of order 6 with all three algorithms. We check whether the models exhibit poles and zeros which are close to each other, leading to almost pole/zero cancellations, or have Hankel singular values



which are several orders of magnitude smaller than the rest, indicating the presence of almost uncontrollable/unobservable states.

Fig. D.9(a) shows the 6 poles along with the 5 zeros of the model obtained by truncating the Loewner matrix pencil to order 6. We notice that the only real zero is close to one of the real poles. Fig. D.9(b) shows the normalized Hankel singular values of this model. Contrary to what we expected, there is no large drop between the last two singular values.

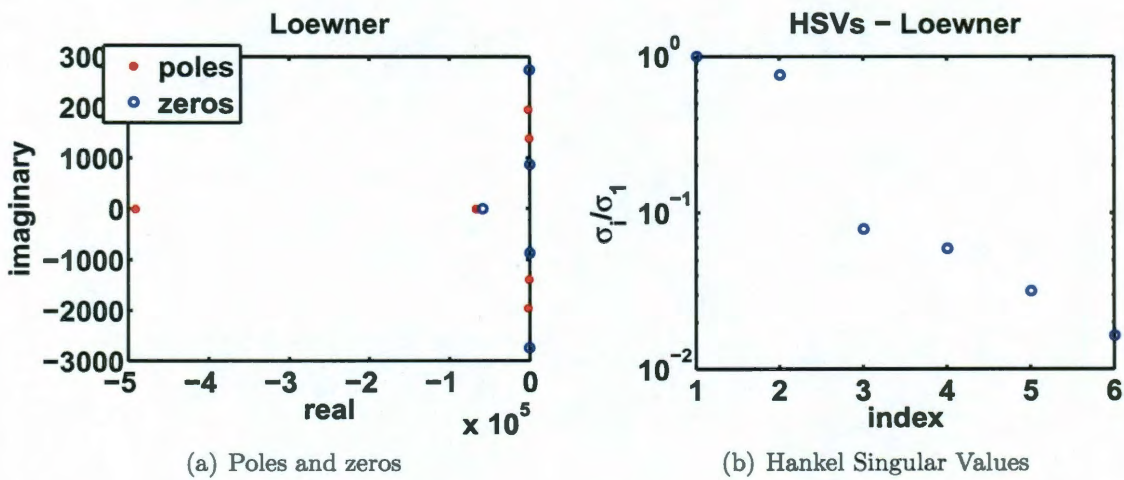


Figure D.9: Results for models of order 6 built with the Loewner framework

Fig. D.10(a) shows the 6 poles along with the 5 zeros of the model obtained with VARPRO. We notice that there is no almost pole-zero cancellation in the real quantities, but rather in two quantities which come in complex conjugate pairs (Fig. D.10(b)). Fig. D.10(c) shows the normalized Hankel singular values of this model, which exhibits no large drop between the last singular values.

Fig. D.11(a) shows the 6 poles along with the 5 zeros of the model obtained with vector fitting. We notice that there is almost pole-zero cancellation in the unstable pole and zero (Fig. D.11(b)). Fig. D.11(c) shows the Hankel singular values of this model normalized with respect to the  $2^{nd}$  singular value, since the first one is infinite as it corresponds to the unstable pole, which exhibits no large drop between the last singular values.

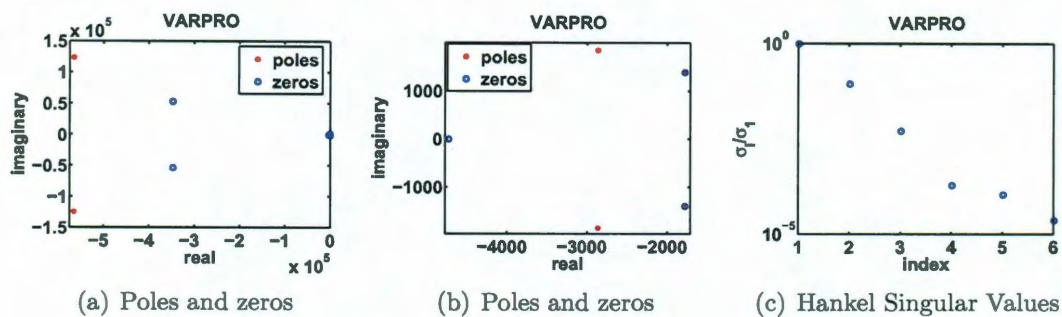


Figure D.10: Results for models of order 6 built with VARPRO

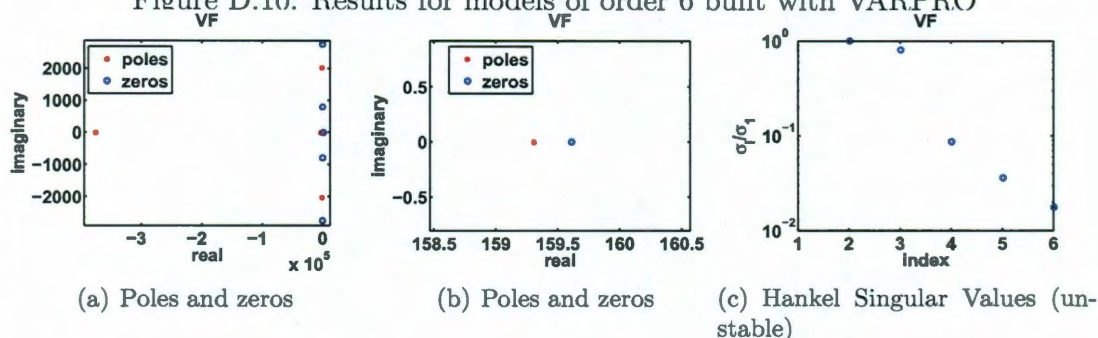


Figure D.11: Results for models of order 6 built with vector fitting

The modeling errors for all three algorithms considered are listed in Table D.7. Except VARPRO, the other two algorithms yield larger errors for this modeling order than for order 5, which was identified as the true model order.

Algorithm	$\mathcal{H}_2$ error	$\mathcal{H}_\infty$ error
Loewner	6.04e-4	9.2e-4
Vector Fitting	9.27e-4	1.6e-3
VARPRO	4.27e-4	6.88e-4

Table D.7: Errors for models of order 6 with  $\mathbf{D} = \mathbf{0}$

We also investigate over-fitting when constructing models of order 15 with all three algorithms.

Fig. D.12(a) shows the 15 poles along with the 14 zeros of the model obtained by truncating the Loewner matrix pencil to order 15. We notice that there are 5 complex poles, together with their complex conjugates, which are located very close to some zeros. Also, these almost pole-zero cancellations occur for poles which are

close to the imaginary axis (in other words, they are low-frequency). Fig. D.12(c) shows the normalized Hankel singular values of this model, which show a large drop between the 5<sup>th</sup> and 6<sup>th</sup> singular values.

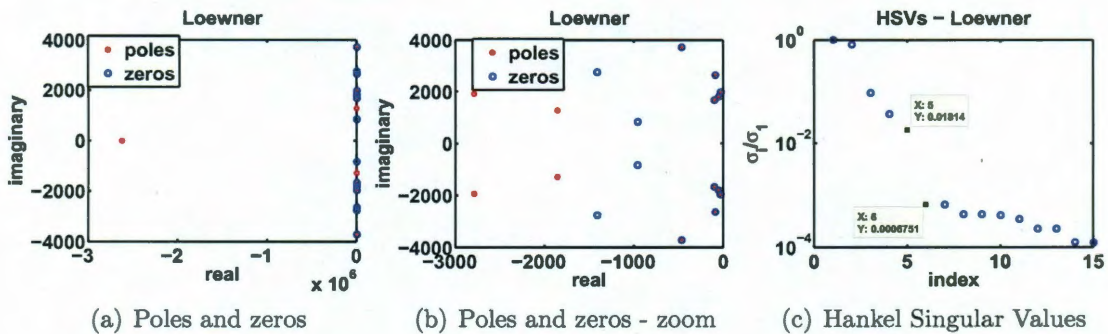


Figure D.12: Results for models of order 15 built with the Loewner framework

Fig. D.13(a) shows the 15 poles along with the 14 zeros of the model obtained with VARPRO. We notice that there is no almost pole-zero cancellation in the real quantities, but rather in two quantities which come in complex conjugate pairs (Fig. D.10(b)). Fig. D.13(c) shows the normalized Hankel singular values of this model, which exhibit a large drop between the 4<sup>th</sup> and 5<sup>th</sup> finite singular values. The first 4 Hankel singular values are infinite due to the 4 unstable poles, so the plot normalizes the finite ones with respect to the largest.

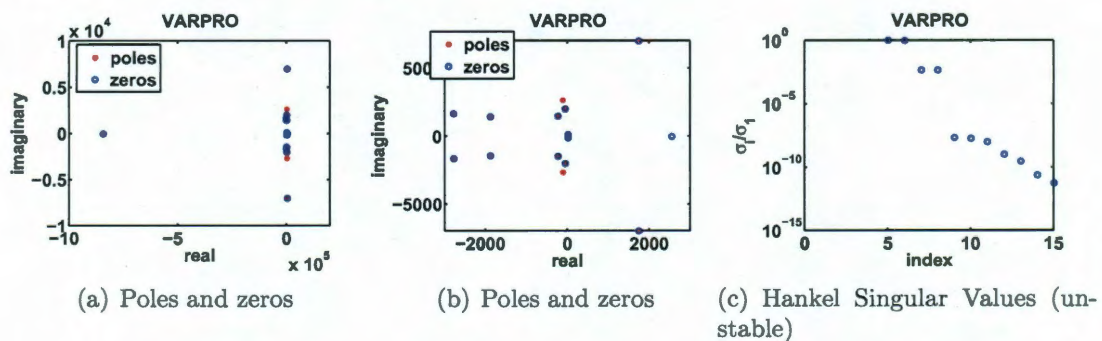


Figure D.13: Results for models of order 15 built with VARPRO

Fig. D.14(a) shows the 15 poles along with the 14 zeros of the model obtained with vector fitting. We notice that there are 10 almost pole-zero cancellations (Fig. D.14(b)). Fig. D.14(c) shows the Hankel singular values of this model normalized

with respect to the 3<sup>rd</sup> singular value, since the first two are infinite as they correspond to the unstable poles. There is a significant drop between the 2<sup>nd</sup> and 3<sup>rd</sup> singular values, as well as between the 9<sup>th</sup> and 10<sup>th</sup>.

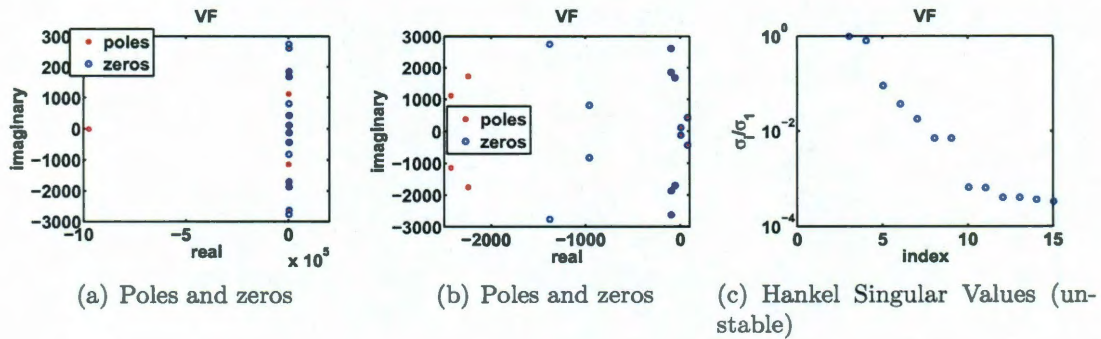


Figure D.14: Results for models of order 15 built with vector fitting

The modeling errors for all three algorithms considered are listed in Table D.8. All algorithms yield smaller errors for this modeling order than for order 5, which was identified as the true model order.

Algorithm	$\mathcal{H}_2$ error	$\mathcal{H}_\infty$ error
Loewner	1.66e-4	4.44e-4
Vector Fitting	2.2e-4	3.99e-4
Varpro	1.43e-4	4.02e-4

Table D.8: Errors for models of order 15 with  $\mathbf{D} = \mathbf{0}$