



**Politecnico  
di Torino**

**ScuDo**

Scuola di Dottorato - Doctoral School  
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation  
Doctoral Program in Electrical, Electronics and Communications Engineering  
(34<sup>th</sup> cycle)

# High-dimensional data-driven parameterized macromodeling

By

**Alessandro Zanco**

\*\*\*\*\*

**Supervisor(s):**

Prof. Stefano Grivet-Talocia

**Doctoral Examination Committee:**

Prof. Mihai Telescu, Referee, Université de Bretagne Occidentale

Prof. Piero Triverio, Referee, University of Toronto

Prof. Marco Storaice, Università degli Studi di Genova

Prof. Guido Lombardi, Politecnico di Torino

Prof. Paolo Manfredi, Politecnico di Torino

Politecnico di Torino

2022

## Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Alessandro Zanco  
2022

\* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

*Dedicated to my grandmother, who made all of this possible*

## Acknowledgements

I would like to express my deepest gratitude to my supervisor Prof. Stefano Grivet-Talocia, for his invaluable support and guidance. I would also express my sincere gratitude to the reviewers, for the effort and the precious comments and insights. Additionally, I could not have undertaken this journey without the invaluable support of the members of the EMC group at Politecnico di Torino, to which goes my profound appreciation.

I would like to extend my sincere thanks to my office mates, for the insightful discussions and moral support.

Lastly, I would be remiss in not mentioning my family, especially my parents, girlfriend and closest friends, whose support and enthusiasm motivated me during this whole journey.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Macromodeling . . . . .	5
1.2	Stability and passivity . . . . .	9
1.3	Objectives . . . . .	9
<b>2</b>	<b>Literature review</b>	<b>12</b>
2.1	Background on macromodeling . . . . .	12
2.1.1	Model order reduction . . . . .	14
2.1.2	Data-driven approach . . . . .	14
2.2	Data-driven macromodeling algorithms . . . . .	15
2.2.1	Model structures for lumped LTI systems . . . . .	15
2.2.2	Fundamental model properties . . . . .	17
2.2.3	The rational approximation problem . . . . .	18
2.2.4	Levy's method . . . . .	19
2.2.5	Generalized Sanathanan-Koerner iteration . . . . .	21
2.2.6	Vector fitting . . . . .	27
2.3	Parameterized macromodeling . . . . .	31
2.3.1	Parameterized macromodeling flows . . . . .	32
2.3.2	Interpolation-based methods . . . . .	32
2.3.3	Parameterized Sanathanan-Koerner iteration . . . . .	40

---

2.3.4	Other techniques . . . . .	53
2.4	Open problems . . . . .	53
<b>3</b>	<b>Passivity of LTI systems</b>	<b>55</b>
3.1	Theoretical aspects of dissipativity . . . . .	57
3.1.1	Dissipative systems . . . . .	57
3.2	Characterization via Linear Matrix Inequalities . . . . .	59
3.3	Passivity characterization in the frequency domain . . . . .	60
3.4	Hamiltonian passivity characterization . . . . .	63
3.5	Enforcing passivity . . . . .	68
3.5.1	LMI constraints . . . . .	69
3.5.2	Hamiltonian perturbation . . . . .	70
3.5.3	Local frequency constraints . . . . .	70
<b>4</b>	<b>Checking and enforcing passivity of parameterized models</b>	<b>73</b>
4.1	Parametric uniform model stability . . . . .	73
4.1.1	Enforcing uniform stability via linear constraints . . . . .	75
4.2	Parametric uniform passivity . . . . .	76
4.2.1	Parameterized Hamiltonian eigenspectrum . . . . .	78
4.3	Perturbation based passivity verification . . . . .	80
4.3.1	Derivatives of SHH pencils and eigenvalues . . . . .	81
4.3.2	Adaptive sampling via Hamiltonian spectral perturbation	84
4.3.3	A relevant test-case . . . . .	89
4.3.4	Comparison with other approaches . . . . .	89
4.4	Enforcing passivity of parameterized models . . . . .	91
4.4.1	Local linearized passivity constraints . . . . .	93
4.4.2	Preserving the accuracy . . . . .	96
4.4.3	Passivity enforcement iteration . . . . .	96

---

4.5	Numerical experiments . . . . .	97
4.5.1	Partially coupled multi-conductor line . . . . .	98
4.5.2	A parameterized multi-board interconnect . . . . .	99
4.5.3	A transmission-line network . . . . .	100
4.5.4	Computational Times . . . . .	103
4.6	Limitations and open problems . . . . .	106
4.7	Adaptive stability enforcement . . . . .	107
<b>5</b>	<b>Towards High-dimensional parameterized macromodeling</b>	<b>114</b>
5.1	Radial Basis Functions approximation . . . . .	115
5.1.1	Scattered data approximation . . . . .	115
5.1.2	Radial Basis Functions . . . . .	115
5.1.3	Mairhuber-Curtis Theorem . . . . .	116
5.1.4	RBF taxonomy . . . . .	118
5.2	High-dimensional macromodeling . . . . .	119
5.2.1	Enforcing uniform stability . . . . .	121
5.2.2	Discussion . . . . .	123
5.3	High-dimensional modeling – experiments . . . . .	124
5.3.1	Extensive performance comparison . . . . .	124
5.3.2	Scalability comparison . . . . .	129
5.4	Discussion . . . . .	134
<b>6</b>	<b>Fully automated macromodeling</b>	<b>136</b>
6.1	Hyper-parameters optimization . . . . .	136
6.2	A sub-optimal shape parameter selection strategy . . . . .	140
6.2.1	Accuracy, conditioning, and shape parameter . . . . .	140
6.2.2	On the numerical conditioning and accuracy . . . . .	144
6.2.3	An algorithm for choosing the shape parameter . . . . .	146

---

6.2.4	From regressor to kernel matrices . . . . .	147
6.2.5	Numerical experiment . . . . .	148
6.3	Choosing RBFs' location and number . . . . .	150
6.3.1	Method 1: concurrent optimization . . . . .	151
6.3.2	Method 2: optimizing only the number of RBFs . . . . .	159
6.4	A fully-automated macromodeling flow . . . . .	163
6.4.1	Optimizing the number of model poles . . . . .	163
6.4.2	An algorithm for automated macromodeling . . . . .	164
6.4.3	Numerical experiments: . . . . .	166
6.5	Passivity for high-dimensional models . . . . .	172
<b>7</b>	<b>Conclusions</b>	<b>176</b>
	<b>References</b>	<b>181</b>
	<b>Appendix A Proof of Theorem 5</b>	<b>193</b>
	<b>Appendix B Proof of Theorem 6</b>	<b>196</b>
B.1	Euclidean Distance Matrices . . . . .	196
B.2	Least Singular Values of Gaussian Kernel Matrices . . . . .	197
B.3	From Kernel to Regressor Matrices . . . . .	199
B.4	Assembling regressor matrices . . . . .	202
	<b>Appendix C Proof of Proposition 1</b>	<b>206</b>
C.1	Proving the upper bound . . . . .	207
C.2	Proving the lower bound . . . . .	207



# Notation

Throughout this manuscript, we will refer with  $\mathbb{N}$ ,  $\mathbb{R}$  and  $\mathbb{C}$  to the natural, real and complex numerical fields, respectively. Scalars are identified with a plain lower-case font  $x$ , vectors with bold lower-case font  $\mathbf{x}$  and matrices with a bold upper-case font  $\mathbf{X}$ . Sets are denoted with a calligraphic notation  $\mathcal{X}$ . The symbol  $j = \sqrt{-1}$  is the imaginary unit and the letter  $s$  is reserved for the complex Laplace variable. The  $i$ -th vector element is identified as  $x_i$ , while the  $(i, j)$ -th matrix element as  $X_{i,j}$ . For any complex number  $z$ ,  $z^*$  denote its complex conjugate. The notation  $\mathbf{X}^\top$  and  $\mathbf{X}^H$  identify the transpose and conjugate transpose (Hermitian) of matrix  $\mathbf{X}$ . Depending on the context, sets are explicitly defined listing their elements between curly brackets  $\{\dots\}$ . The symbols  $\mathbf{I}_P$ ,  $\mathbf{1}_{m,n}$  and  $\mathbf{1}_m$  are reserved, respectively, for the identity matrix of dimension  $P$ , the  $m \times n$  matrix and the vector of dimension  $m$  with all elements equal to one.. Dimensions will be omitted when clear from the context. The operators  $\det(\mathbf{X})$  and  $\text{rank}(\mathbf{X})$  evaluate the determinant and the rank of their matrix arguments. The notation  $\lambda(\mathbf{X})$  and  $\sigma(\mathbf{X})$  denote the eigenvalue and singular value spectra, respectively. Depending on the context, the operator  $\text{eig}(\mathbf{X})$  is also used to identify the eigenvalues of matrix  $\mathbf{X}$ . Vector and (induced) matrix  $p$ -norms are denoted as  $\|\mathbf{x}\|_p$  and  $\|\mathbf{X}\|_p$ ; when  $p$  is omitted, we refer to the standard Euclidean 2-norm.

# Summary

Mathematical modeling has become essential for the electronic industry. Today's market necessities demand to match the conflicting requirements of increasingly faster workflows, together with reliable and accurate designs. Assessing the robustness of designs may be extremely challenging, due to the presence of increasingly miniaturized devices, in conjunction with high-frequency signals and other interfering components placed in close proximity to each other. In this context, modeling the impact of distributed effects on the reliability of complex designs demands for sophisticated simulation software that, solving first-principle equations (e.g., Maxwell's equations), are capable of accurately predicting their electrical and electromagnetic behaviors.

However, the impressive accuracy levels achieved by first-principle solvers come at the cost of prohibitively long runtimes, that badly fit the above-mentioned design requirements. In this scope, the use of simplified reduced-order behavioral models (or, macromodels) is extremely helpful. Retaining only a reduced set of auxiliary variables necessary to accurately reproduce the input/output behavior or interest, macromodels can replace the original fully detailed description, enabling for exceptionally fast simulations with a minimal and controlled error.

The possibility of embedding in the macromodel the variability upon some parameters, would additionally enable cheap repeated simulations under different configurations of the structure of interest and different environmental conditions, allowing for variability, optimization and design centering analyses that would be impractical with first-principle solvers. This manuscript fits in this last parameterized framework where, although well-consolidated parameterized macromodeling strategies are already available, major open problems are still to be addressed. Throughout this work we will consider data-driven

approaches, so that parameterized models are constructed from a reduced set of frequency-domain data, available from simulation.

First, we address the problem of certifying the model stability and passivity, which are fundamental properties that the model must have to be of practical use in design flows. A first set of approaches propose to build parameterized models that are stable and passive by construction, at the cost of a reduced portability and lack of compactness. Others, instead, enforce these properties employing a post-processing perturbation strategy, guaranteeing the model compactness. In this second framework, prior to this work only partial solutions were available, that were limited to one parameter only. The first technical contribution of this work is an extension of the latter approach to higher-dimensional parameter spaces. Formulating a parameter-dependent Hamiltonian matrix, we set up a scalable algorithm for the identification of stability and passivity violations, to be removed in a subsequent post-processing phase.

Then, we address the problem of *high-dimensional* macromodeling, i.e., the inclusion of a much larger number of parameters in the model, that would be much more flexible since representative of a wider set of design and working conditions. Standard parameterized macromodeling strategies are not suited for high-dimensional tasks, since their identification procedures become exponentially more complex as the number of parameters increases. The second technical contribution of this work is the formulation of a high-dimensional model form based on *Radial Basis Functions*, that is specifically suited to handle many more parameters without incurring in computational issues. In addition, we also introduce an innovative set of uniform stability constraints, whose formulation does not depend on the number of parameters and, thus, perfectly fits the considered high-dimensional setting.

The last contribution of this thesis is the formulation of a comprehensive set of strategies, specifically aimed at optimizing, based on the dataset at hand, the dimension of the approximation spaces and the hyper-parameters of the radial basis functions, basically without any user interaction. This enables the development of a high-dimensional fully automated macromodeling flow that autonomously optimizes the model structure and generates accurate high-dimensional and guaranteed stable models.

In summary, this dissertation provides advancements under both a theoretical and a practical viewpoint. The improvements with respect to prior research works are documented with relevant numerical examples, demonstrating that the results of this work are ready for exploitation.

# Chapter 1

## Introduction

The impressive miniaturization achieved in the last decades by electronic systems, together with the presence of high-frequency signals and the increasingly shorter time-to-market, demand for extremely robust and effective product design phases. In this view, Electronic Design Automation (EDA) tools emerge as essential assets for the engineers, who can validate the performances of each individual design step through repeated numerical simulations, rather than resorting to expensive prototyping setups. Most often, simulations are performed via dedicated software that, based on first-principle equations (in our case, the Maxwell's equations), predict the electrical and electromagnetic behavior of the device under test. In order to accurately estimate the few electrical and/or electromagnetic variable of interest, the first-principle solvers necessitate to compute millions (or even billions) of intermediate “internal” unknowns. Hence, the high-accuracy simulation results achieved by general purpose software may be extremely demanding in terms of computational resources, to the point of requiring very expensive ad-hoc hardware configurations.

### 1.1 Macromodeling

From these considerations stems the push toward *Reduced Order Models* (ROMs) or *macromodels* that, at the cost of a negligible and controlled error, allow predicting the variables of interest with a minimum effort. This is made possible by reducing the complexity of the highly detailed solution provided by first-

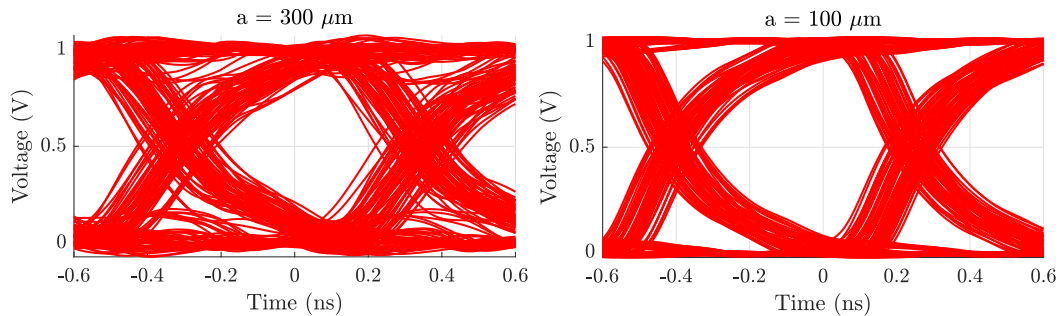


Fig. 1.1 Eye diagram analysis of an high-speed interconnect (see [3]) obtained with a SPICE engine for two different values of the parameter  $a$  (the multi-layer PCB via radius). The simulation is performed by injecting a pseudo-random bit sequence. Reproduced with permission from [4], © 2017 IEEE

principle solvers, in order to retain only a reduced set of subsidiary variables that accurately reproduce the input-output relations of interest. Therefore, ROMs can be successfully employed, for instance, as surrogates of more complex circuit blocks in time-domain simulations, or as part of more complex software suites, where they are often employed in the intermediate simulation steps to improve their performances. The strong demand for electrical and electromagnetic macromodeling tools pushed for the development of commercial solutions, such as IdEM [1] and BroadBand Spice [2].

The possibility of extending the above macromodeling scheme to mimic the input-output responses of interest under different configurations of design and external environmental parameters (e.g., geometrical dimensions, components values, incident EM fields, etc.) can be of great advantage for the final design centering and optimization steps (e.g., eye-diagram analyses of Figure 1.1). Indeed, these are often performed through Monte Carlo runs that, requiring repeated solver calls (each corresponding to a different combination of the parameters), would require impractically long runtimes even with sophisticated hardware. Conversely, parameterized macromodels are perfectly suited for these purposes.

## An example

Let us make these statements clearer with an example. Suppose to be given with the RF matching network (first presented in [5]) depicted in Figure 1.2,

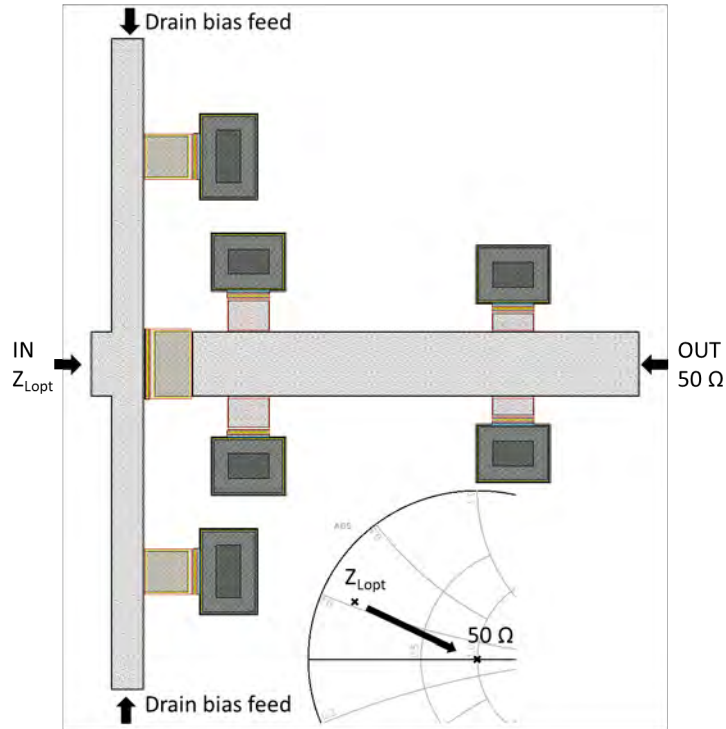


Fig. 1.2 Layout of the considered RF matching network. From [5] © 2022 IEEE

used to match the output impedance ( $5.8 + j10.9 \Omega$ ) of a large-periphery GaN transistor operating at 28 GHz to the reference impedance of  $50 \Omega$ . The multi-step semi-lumped pi-type matching network includes a short-circuited stub (also used for device biasing), large series DC-block capacitor and two shunt capacitors acting as open-circuited stubs. All the components have been split in parallel pairs. Large and small MIM capacitors have been used in the RF-short and DC-block, and as open-circuited stubs, respectively. Large MIM capacitors are implemented by means of a silicon nitride layer ( $\approx 150$  nm), while small capacitors are realized with an additional oxide layer ( $\approx 800$  nm). According to the statistical data, a reasonable relative standard deviation for the layer thicknesses is assumed to be 5% and 3%, respectively.

In order to assess the performances of the matching network under different configurations of the layer thicknesses, we run a Monte Carlo analysis with 500 trials, assuming uncorrelated variations of the parameters. All the subsequent simulations have been carried out with Keysight ADS [6].

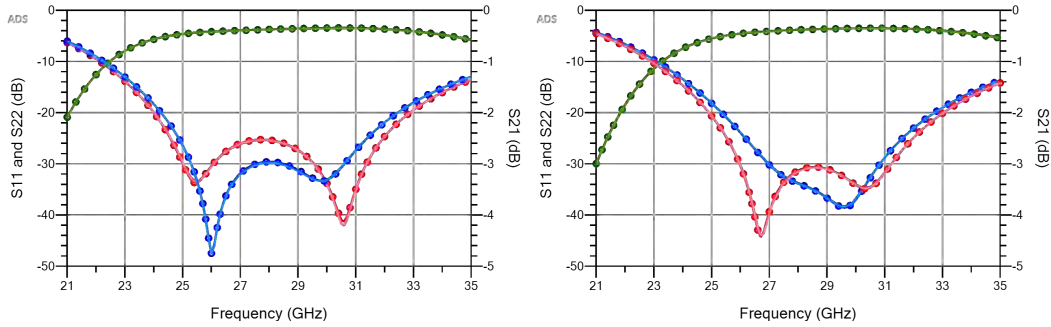


Fig. 1.3 Model responses (dots) compared with the EM solver data (light solid lines) for the scattering matrix elements  $\mathbf{S}_{1,1}$  (red),  $\mathbf{S}_{2,1}$  (green) and  $\mathbf{S}_{2,2}$  (blue). Left panel: nominal parameter configuration. Right panel: perturbed parameter configuration. From [5] © 2022 IEEE

We first run full EM simulations with 200 frequency samples. In this setting, each solver run requires approximately 50 minutes, so that the full Monte Carlo would require 17 days. Of course, this is impractical.

Then, exploiting some optimizations embedded in the EM solver we have been able to complete each parametric simulation in about 3 minutes, corresponding to 25 hours for the complete Monte Carlo run.

Let us now run the same variability analysis with a parametric macromodel. The first step is to derive a reduced yet significant set of initial parametric scattering responses at the network ports. For this example, we precomputed (employing the optimized version of the ADS Momentum EM solver) 9 parametric frequency responses, for a total CPU time of approximately 30 minutes. These data are subsequently used to train the model by means of ad-hoc identification algorithms, that will be discussed in Chapter 2. As depicted in Figure 1.3 for the nominal (left panel) and for a perturbed (right panel) parameter configuration, the model (dots) accurately reproduce the original data (light solid lines).

Then, the parameterized model is converted into an equivalent SPICE netlist (see Chapter 2 for details), that is used in ADS in place of the original complex matching network. In this setting, the Monte Carlo analysis took less than a minute. Thus, considering the additional overhead required to evaluate the training responses ( $\approx 30$  minutes) and the model extraction that took few seconds, using a macromodel enables for a  $48\times$  speedup with an error in



the order of  $10^{-3}$  with respect to the original EM solver data. However, once the macromodel is available, it is meant to be reused multiple times without repeating the identification procedure. Hence, considering only the simulation time, we achieve a speedup of approximately  $1500\times$  when compared with the optimized simulation. This is not an isolated case. Indeed, compressions in the order of approximately  $1000\times$ , corresponding to approximation errors below 1% are common when using macromodels.

## 1.2 Stability and passivity

Beside the accuracy, in order to be reliably employed as surrogates in time-domain simulations, macromodels must also be representative of some physical properties of the reference circuit block, namely, stability and passivity. Models that are not stable or not passive may be the root cause of unreliable simulations and must be avoided. We refer the Reader to Chapter 3 for a detailed discussion on this topic, including a set of relevant examples.

Guaranteeing stability and passivity of parameterized models has been tackled under several standpoints. Some techniques [7–9] propose a model structure that is stable and passive by construction, at the cost of a reduced model compactness and portability. Others [10, 11], instead, generate compact parameterized models that may require some post-processing manipulations to enforce stability and passivity. A comprehensive review is available in Chapter 2.

Schemes based on stability and passivity enforcement as a post-processing, in particular the *Parameterized Sanathanan Koerner* (PSK) approach, will be the algorithms of choice throughout this work. The corresponding data-driven macromodeling flow that we will consider in this work is illustrated in Figure 1.4.

## 1.3 Objectives

One of the main objectives of this work is the definition of a set of model structures, constraints, and numerical algorithms for enriching parameterized

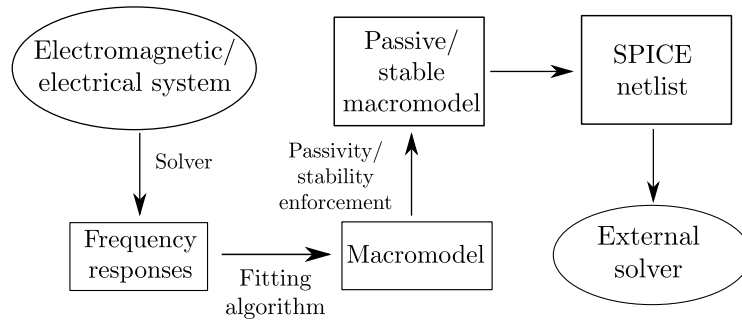


Fig. 1.4 Passive data-driven macromodeling flow adopted in this work.

macromodeling tools in order to ensure model stability and passivity. Considering the model transfer function as a multivariate function of frequency and one or more external parameters, the goal is then to make the model poles uniformly stable throughout the parameter space, as well as to make the model dissipative, again throughout the parameter space. No definitive solution to this problem exists in the general case. Our proposed solutions are presented in Chapter 4.

A common limitation, shared among *all* the available parameterized data-driven macromodeling schemes, is the impossibility of embedding more than few independent parameters, due to an exponentially increasing model complexity, which results in impractically demanding identification procedures. In this work we will also address this problem, by proposing an innovative *high-dimensional* parameterized model structure built on *Radial Basis Functions* (RBF). This is the subject of Chapter 5 where, in addition, we will also introduce innovative stability constraints whose formulation is independent of the number of parameters, making them perfectly suited for high-dimensional models.

Besides the excellent scalability properties of the proposed RBF schemes, good approximation results come at the cost of finely tuning some hyper-parameters of the RBFs. In literature, the problem of estimating optimal values for the hyper-parameters is well-known [12–15]. As we will thoroughly discuss in Chapter 6, a direct application of these schemes is not feasible. In Chapter 6 we introduce a set of innovative hyper-parameter optimization techniques, specifically tailored to our setting, that allow for an extremely quick and effective estimation of sub-optimal hyper-parameter values.

---

Building upon the above techniques in this manuscript, the final result of this work summarized in Chapter 6 is the formulation of a high-dimensional macromodeling flow, that enables the efficient construction of guaranteed stable parameterized macromodels with less to none user interaction, thus filling a relevant gap in the literature. More practically, industries would benefit from the results of this work in terms of more efficient and productive EDA-based design flows, in all those cases where a fast design space exploration, optimization, sensitivity and Monte Carlo analyses are required. Ultimately, simulation software vendors may take advantage of the proposed automated macromodeling methods to improve the performances of their commercial products.

# Chapter 2

## Literature review

### 2.1 Background on macromodeling

With the term *macromodel* we refer to a mathematical description of some device which is primarily aimed at accurately reproducing its input-output *behavior*. For this reason, this approach is also often referred to as *behavioral modeling*. Within this paradigm, we can identify two major and complementary strategies, namely the *white-box* and *black-box* approaches.

The *white-box* setting assumes some degree of knowledge about the device we are modeling (e.g., the topology of a lumped element electrical circuit). This information is used to construct ad-hoc model forms that are representative of the device of interest. Depending on the problem at hand, the translation of physical knowledge into a model requires some degree of approximation that must be kept under control to maximize the model accuracy. Often, the modeling step involves the optimization of the model parameters (e.g., the values of lumped components) and/or the model structure (e.g., augmentation of a basic circuit topology [16]) based on the available raw data.

The main advantage of this approach comes with the intrinsic *interpretability* of the model that, for instance, can be extremely useful to designers due to the strong correlation between the model and the real-world device. In addition, enforcing some constraints on the component values, it is possible to easily guarantee some physical and/or structural properties that the model must reproduce (e.g., passivity, stability).

This approach finds its limits when dealing with moderately to highly complex problems (e.g., distributed networks with non-negligible propagation time, large interconnect networks, etc...). In those cases, the problem is twofold. First, we may be limited in the knowledge of the device we are modeling, which thus translates to incorrect or unreliable model structures. In addition, for an increasingly large number of components, the complexity of the resulting optimization problem (often non-convex and characterized by many local minima) increases, becoming easily intractable.

On the other side, we denote with the term *black-box* a model defined only by the functional relations between the input/output ports, disregarding any prior knowledge about the device under modeling. The main advantage of the black-box approach is its flexibility, which stems from a model structure that, instead of being case specific (as in the white-box approach), is purposely intended to be more general. This enables to model more complex phenomena such as, for instance, strong high-frequency effects (skin and proximity effect) or electromagnetic couplings that affect the device of interest.

Moreover, a black-box model can be safely shared without disclosing proprietary information about sensitive device characteristics, as it retains only the input/output relations. This is a great benefit for vendors, who can safely share component models with their customers. This, of course, comes with a reduced interpretability of the model.

The absence of physical-based constraints in the model description may lead to models lacking physical consistency. For instance, characteristics such as stability and passivity may not be guaranteed by construction and thus must be enforced with ad-hoc techniques [17, 18]). Enforcing uniform stability and passivity in the parameterized setting is one of the key challenges addressed in this work (see Sections 4.4, 4.7).

Throughout this work, we will focus on the category of black-box macromodels. Hence, with the term *macromodel* we will implicitly refer to a *black-box* behavioral model. For the interested Reader, we refer to [19] for additional details on white-box modeling.

Depending on the steps that lead to a macromodel, we can identify two main approaches.

### 2.1.1 Model order reduction

Often, an accurate description of the phenomena of interest requires handling very large systems of (possibly coupled) differential equations. This primary set of equations may be derived, for instance, from discretization of field equations or from a *Modified Nodal Analysis* (MNA) description of a lumped network. Usually, such a kind of detailed electrical or electromagnetic characterization takes into account both the main functional input/output characteristics and for all the capacitive/inductive or distributed parasitic effects. In the scenario of numerical simulation, the solution of these highly complex models turns out to be extremely demanding both in terms of memory and CPU requirements.

A Model Order Reduction (MOR) approach *reduces* the complexity of the models (*compression*) while, at the same time, approximating the input-output characteristics – the larger is the compression, the less accurate is the Reduced Order Model (ROM).

Some of the most common MOR approaches rely on *projections* of an original large-sized state-space description into lower dimensional sub-spaces, eliminating possible redundant (thus unnecessary from an input/output standpoint) dynamics. The MOR algorithm PRIMA (*Passive Reduced-Order Interconnect Macromodeling Algorithm* [20]), is one of the most successful due to its passivity preserving and controlled accuracy properties.

Other techniques, known as *truncation* methods, reduce the model complexity eliminating some system dynamics with small contributions (measured in some suitable norm) to the overall input/output characteristics. The most common truncation method is known in the literature as *Truncated Balanced Realization* (TBR) [21, 22].

For a more detailed description of Model Order Reduction techniques, we refer the Reader to [23–25]

### 2.1.2 Data-driven approach

The Model Order Reduction approach requires, as a starting point, a detailed description of the structure of interest. This information is often available in proprietary and ad-hoc software, but is hidden to the user in common

commercial products that provide only the input-output results. Another common scenario where an accurate mathematical characterization of the device can not be derived, arise in all those cases of measurements performed on physical hardware. The data acquired from the experiments may come in the form of tabulated scattering responses obtained by means of Vector Network Analyzers (VNA) or transient responses evaluated at predefined input-output port combinations.

A macromodeling approach based solely on a sampled input/output characterization, either simulated or experimental, is defined *data-driven* and, since no internal description is required, can be considered fully black-box.

A data-driven macromodeling flow can be split in three separate and subsequent steps. First, the set of input/output data must be computed (or measured) in correspondence to predefined  $P$  electrical interface ports. Then, a curve fitting algorithm is used to synthesize an accurate macromodel representative of the given raw data. As a final step, if necessary, the macromodel is suitably post-processed to enforce possible physical inconsistencies (i.e., stability and passivity).

This work will focus only on the category of *data-driven* macromodeling performed in the frequency domain. For the Reader who is interested, we report some key references concerning time domain approaches [26–28].

## 2.2 Data-driven macromodeling algorithms

In this Section, we are going to review some state-of-the-art algorithms that are widely adopted in the context of data-driven modeling of linear systems.

### 2.2.1 Model structures for lumped LTI systems

Often, general purpose model structures (e.g., neural networks) are the go-to choice when the inner workings of the device under modeling are completely unknown. This approach is too general for the objectives of this work, as we know that the structures of our interest can be accurately described by Linear-Time-Invariant (LTI) models, thus in terms of coupled Ordinary Differential

Equations (ODEs). The (possibly) high complexity of these systems of linear differential equations calls for a representation that links the (time-dependent) input and output vectors  $\mathbf{u}(t) \in \mathbb{R}^P$  and  $\mathbf{y}(t) \in \mathbb{R}^P$  in a compact and meaningful way. In this view, it is common to use the so-called *state-space* realization of the differential equation, defined as (the shorthand notation  $\dot{\mathbf{x}}$  refers to the first time-derivative of vector  $\mathbf{x}$ )

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \quad (2.1)$$

The matrices  $\mathbf{A} \in \mathbb{R}^{\bar{N},\bar{N}}$ ,  $\mathbf{B} \in \mathbb{R}^{\bar{N},P}$ ,  $\mathbf{C} \in \mathbb{R}^{P,\bar{N}}$ ,  $\mathbf{D} \in \mathbb{R}^{P,P}$  are called *realization matrices* and the time-dependent vector  $\mathbf{x}(t) \in \mathbb{R}^{\bar{N}}$  collects the model *states*. In those cases where the input/output relations are governed also by purely algebraic equations (in addition to the ODEs), we generalize the state-space realization to the so-called *descriptor* realization, that reads

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \quad (2.2)$$

where the matrix  $\mathbf{E} \in \mathbb{R}^{\bar{N},\bar{N}}$  can be singular. For further use, we introduce the following definitions [29]:

- a descriptor system is regular if  $\exists s \in \mathbb{C}$  such that  $\det(s\mathbf{E} - \mathbf{A}) \neq 0$ .
- a descriptor system is impulse-free if  $\text{rank}(\mathbf{E}) = \text{deg}(\det(s\mathbf{E} - \mathbf{A}))$ .

where the operator  $\text{deg}(\cdot)$  returns the degree of its polynomial argument.

The descriptor realization is more flexible if compared with the standard state-space; as an example, the MNA equations can be cast in descriptor form and not in the standard state-space.

In the following, we will refer to the above realizations with the shorthand notation

$$\underbrace{\left( \begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{array} \right)}_{\text{State-space}}, \quad \underbrace{\left( \begin{array}{c|c} [\mathbf{A}, \mathbf{E}] & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{array} \right)}_{\text{Descriptor}} \quad (2.3)$$



Beside the above time-domain description based on ODEs, thanks to the linearity assumption, it is possible to equivalently describe the input/output relation of interest in the frequency domain. In this view, we refer to the ratio

$$\mathbf{H}(s) = \frac{\mathbf{Y}(s)}{\mathbf{U}(s)} \in \mathbb{C}^{P,P} \quad (2.4)$$

as *transfer function*, where  $s = \sigma + j\omega$  is the Laplace variable and  $\mathbf{Y}(s) \in \mathbb{C}^P$ ,  $\mathbf{U}(s) \in \mathbb{C}^P$  denote the Laplace transforms of the output  $\mathbf{y}(t)$  and input  $\mathbf{u}(t)$ , respectively. The following well-known [19] relation between the state-space (2.1) and descriptor (2.2) realizations hold

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \quad \text{for state-space realizations,} \quad (2.5)$$

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \quad \text{for descriptor realizations} \quad (2.6)$$

Regular state-space realizations (2.5) are impulse-free by construction, whereas descriptor realizations (2.6) are impulse-free when  $\mathbf{H}(\infty)$  is bounded.

In our data-driven macromodeling framework, the model  $\mathbf{H}(s)$  should reproduce the broadband behavior of the structure under modeling, characterized by the  $P^2$  responses collected as entries in a data matrix  $\check{\mathbf{H}}(s)$ .

## 2.2.2 Fundamental model properties

Together with the accuracy with respect to the available raw data  $\check{\mathbf{H}}(s)$ , we additionally require our models to be representative of physically consistent electrical and electromagnetic phenomena. In particular, we refer to the concepts of *stability* and *passivity*.

### Stability

A LTI system described by (2.1) or (2.2) is physically consistent only if its states  $\mathbf{x}(t)$  are bounded for any considered time instant  $t$ . We refer to this property as to *stability*. In particular, in this work we will consider the stronger definition of *asymptotic stability*, which requires the states  $\mathbf{x}(t)$  of the autonomous (non-excited) system  $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$  to vanish as  $\mathbf{x}(t) \rightarrow \mathbf{0}$  for  $t \rightarrow \infty$ , for any finite initial condition  $\mathbf{x}(0)$ . A LTI system is asymptotically stable if the  $\bar{N}$

eigenvalues  $\lambda_i$  of matrix  $\mathbf{A}$  in (2.1) (resp. the finite eigenvalues of the pencil  $(\mathbf{A}, \mathbf{E})$  in (2.2)) have a strictly negative real part [30]. Equivalently, a LTI system described in the frequency domain by the transfer matrix  $\mathbf{H}(s)$  is asymptotically stable if  $\mathbf{H}(s)$  does not have singularities in the closed right-half complex plane.

### Passivity

In addition to the above internal description, the concept of passivity also considers the input/output net energy flow. In particular, we define as *passive* any structure that is unable to generate energy by its own. Examples of passive devices are ubiquitous, ranging from simple resistors to more complex structures such as, for instance, Printed Circuit Boards (PCBs). Macromodels of passive devices must reproduce this property when used as black-box surrogates in time-domain simulations; otherwise, the nonphysical energy injected in the network by the non-passive model may drive the whole simulation to instability (see Figure 3.1 in Chapter 3).

It has been proven that a LTI model of the form (2.1) or (2.2) is passive if it satisfies the *Positive Real Lemma* (PRL) or the *Bounded Real Lemma* (BRL) for the immittance and scattering representation, respectively. More details on these properties and a set of relevant results will be provided in the dedicated Chapter 3.

### 2.2.3 The rational approximation problem

For simplicity, in the following we will refer to the scalar *Single Input - Single Output* (SISO) case (i.e.,  $P = 1$ ). Extension to the general matrix-valued *Multi Input - Multi Output* (MIMO) case (i.e.,  $P > 1$ ) will be discussed later, starting from Section 2.2.5.

For a generic LTI one-port (SISO) system, the transfer function  $H(s) \in \mathbb{C}$  comes as a rational function of  $s$ , as

$$H(s) = \frac{N(s)}{D(s)} = \frac{a_{\bar{n}}s^{\bar{n}} + a_{\bar{n}-1}s^{\bar{n}-1} + \dots + a_0}{b_{\bar{m}}s^{\bar{m}} + b_{\bar{m}-1}s^{\bar{m}-1} + \dots + b_0} \quad (2.7)$$

where  $\bar{n}$ ,  $\bar{m}$  denote the orders of numerator and denominator, respectively. From now on, it is assumed that the orders  $m$  and  $n$  are equal, so that  $H(s)$  is a proper rational function. This choice implies the model transfer function to be asymptotically bounded, which is suitable to approximate, for instance, the scattering parameters of an electromagnetic device. Extensions to the general case  $m \neq n$  are straightforward, more details can be found in [19].

We assume that the data we want to model come in the form of tabulated frequency responses<sup>1</sup>  $\check{H}(s_k) \in \mathbb{C}^{P,P}$  (for instance, scattering parameters), evaluated in correspondence of discrete frequency samples  $s_k = j\omega$ ,  $k = 1, \dots, K$ . Our objective is to find suitable coefficients  $a_n, b_n$ , such that the model (2.7) approximates accurately the available data, as

$$H(s_k) \approx \check{H}(s_k), \quad \text{for } k = 1, \dots, K \quad (2.8)$$

Defining the approximation error  $e_k = e(s_k)$  as

$$e_k = H(s_k) - \check{H}(s_k) = \frac{N(s_k)}{D(s_k)} - \check{H}(s_k) \quad (2.9)$$

we collect all these elements in vector  $\mathbf{e}$  and introduce the cumulative error metric  $\|\mathbf{e}\|_2$ . The best approximation is reached for the optimal set of coefficients  $\mathbf{c} = [a_0, \dots, a_{\bar{n}-1}, a_{\bar{n}}, b_0, \dots, b_{\bar{n}-1}, b_{\bar{n}}]^T$  that satisfies

$$\mathbf{c}_{opt} = \arg \min_{\mathbf{c}} \|\mathbf{e}\|_2 \quad (2.10)$$

The rational optimization problem (2.10) is non-linear and non-convex, since the unknowns  $b_0, \dots, b_{\bar{n}}$  appear at the denominator. In general, the solution of rational optimization problems is particularly problematic as the cost-function is non-convex and may be characterized by multiple local minima that, in turn, may be far from the global optimum.

### 2.2.4 Levy's method

Historically, one of the first attempts to recast the non-convex optimization problem (2.8) into a more manageable form goes back to the work from E.

<sup>1</sup>These are often stored in standard `.snp` Touchstone format

C. Levy [31] published in 1959. Provided that the non-convexity arises from the presence of decision variables appearing at the denominator of  $H(s)$ , the main idea is to reformulate the cost function, eliminating the unknowns at the denominator. Hence, in [31] Levy proposes a modified error metric  $\mathbf{e}^L$  whose elements  $e_k^L$  are obtained by multiplying the elements  $e_k$  by the corresponding denominator  $D(s_k)$

$$e_k^L = D(s_k)e_k = N(s_k) - D(s_k)\check{H}(s_k) \quad (2.11)$$

The modified Levy's error  $\mathbf{e}^L$  becomes thus linear in the decision variables, and the resulting optimization problem

$$\arg \min_{\mathbf{c}} \|\mathbf{e}^L\|_2 \quad (2.12)$$

can be easily cast in least-squares form, as

$$\arg \min_{\mathbf{c}} \|\Psi\mathbf{c}\|_2 \quad (2.13)$$

where  $\Psi = (\Phi, -\check{\mathbf{H}}\Phi)$ ,  $\check{\mathbf{H}} = \text{diag}\{\check{H}(s_1), \dots, \check{H}(s_K)\}$  and

$$\Phi = \begin{pmatrix} 1 & s_1 & \dots & s_1^{\bar{n}} \\ \vdots & & & \vdots \\ 1 & s_K & \dots & s_K^{\bar{n}} \end{pmatrix}, \quad (2.14)$$

The trivial vanishing solution  $\mathbf{c}_{opt} = \mathbf{0}$  of (2.13) is avoided by suitably constraining the optimization problem.

There are two main problems that affect this approach:

1. The linearization introduced in (2.11) introduces a frequency-dependent bias in the solution [19].
2. If the poles span several decades, the term  $|D(s_k)|^2$  is forced to attain widely varying values when evaluated on the experimental samples  $s_k$ . The frequency-dependent bias affecting the solution may thus lead to loss of accuracy over the fitting band.

3. The Vandermonde-like structure of the matrix  $\Psi$  lead the optimization problem to be numerically unstable when the polynomial degree  $\bar{n}$  exceed few units.

### 2.2.5 Generalized Sanathanan-Koerner iteration

The above problems has been successfully tackled by the so-called *Sanathanan-Koerner* (SK) algorithm, then by its extension, the *Generalized Sanathanan-Koerner* (GSK) iteration. Since this formulation, as well as the subsequent *vector fitting* algorithm, are key for later developments, a detailed review is reported below, in order to set notation and properly establish background results. For the same reason we will consider the general MIMO case.

Let us consider the following rational model,

$$\mathbf{H}(s) = \frac{\mathbf{N}(s)}{D(s)} = \frac{\mathbf{R}_{\bar{n}}s^{\bar{n}} + \mathbf{R}_{\bar{n}-1}s^{\bar{n}-1} + \dots + \mathbf{R}_0}{b_{\bar{n}}s^{\bar{n}} + b_{\bar{n}-1}s^{\bar{n}-1} + \dots + b_0} \quad (2.15)$$

where the numerator matrix coefficients  $\mathbf{R}_n$  are responsible for the multi-input multi-output structure. A common denominator among all transfer function elements is compatible with (2.5) and (2.6).

We start addressing the problem of the frequency dependent bias, whose complete compensation demands to divide the linearized (matrix valued) Levy's residual  $\mathbf{E}_k^L$  by the denominator, i.e.,

$$\frac{\mathbf{E}_k^L}{D(s_k)} = \frac{\mathbf{N}(s_k) - D(s_k)\check{\mathbf{H}}(s_k)}{D(s_k)} \quad (2.16)$$

that we observed earlier to be non-convex in the decision variables. Instead of aiming for a perfect bias compensation, the approach advised by Sanathanan and Koerner in [32] starts from (2.16) and proposes a relaxed iterative scheme that, at each iteration  $\nu$ , involves the solution of a *linear* least-squares that minimizes the modified residual

$$\mathbf{E}_{k,\nu}^{SK} = \frac{\mathbf{N}_{\nu}(s_k) - D_{\nu}(s_k)\check{\mathbf{H}}(s_k)}{D_{\nu-1}(s_k)} \quad (2.17)$$

Differently from the original non-convex formulation (2.16), in (2.17) the denominator  $D_{\nu-1}(s_k)$  is known at each subsequent iteration  $\nu$ , leading the fitting residual (2.17) to be linear in the decision variables. At convergence, the denominator estimates stabilize, i.e.,  $D_\nu(s) = D_{\nu-1}(s)$ , hence

$$\mathbf{E}_{k,\nu}^{SK} \xrightarrow{D_\nu=D_{\nu-1}} \frac{\mathbf{N}_\nu(s_k)}{D_\nu(s_k)} - \check{\mathbf{H}}(s_k) = \mathbf{E}_k \quad (2.18)$$

where  $\mathbf{E}_k$  corresponds to the original unbiased error functional. Therefore, the weighting introduced in dividing by  $D_{\nu-1}(s_k)$  iteratively removes the bias arising from the linearization. In literature, this is known as *Sanathanan-Koerner* (SK) algorithm.

In turn, the compensation of the frequency-dependent bias attained by the SK iteration guarantees a controlled error over the entire frequency band, independently of the number and distribution of the poles. This enables to extend the applicability of such a rational approximation scheme to electromagnetic structures characterized by poles spanning several decades (e.g., transmission lines, distributed networks, etc. . . ).

We now address the numerical issues inherently associated with the adopted model rational functional form. It is indeed well known that polynomial approximation problems may be subject to numerical issues and ill-conditioning, especially in case of large polynomial orders. We should therefore find a more suitable model structure that leads to better conditioned optimization problems. In general, both numerator and denominator can be expressed in terms of linear combinations of some prescribed basis functions  $\varphi_n(s)$ , resulting in

$$\mathbf{H}(s) = \frac{\mathbf{N}(s)}{D(s)} = \frac{\sum_{n=0}^{\bar{n}} \mathbf{R}_n \varphi_n(s)}{\sum_{n=0}^{\bar{n}} b_n \varphi_n(s)} \quad (2.19)$$

An appropriate selection of basis functions  $\varphi_n(s)$  should take into account the following constraints:

1. the model  $\mathbf{H}(s)$  must be a rational function in the variable  $s$ ,
2. the optimization problem should be well-conditioned for any choice of  $\bar{n}$ .

These two constraints are concurrently fulfilled setting the basis functions as partial fractions, defined as

$$\begin{cases} \varphi_n(s) = 1, & n = 0 \\ \varphi_n(s) = \frac{1}{s-q_n}, & n = 1, \dots, \bar{n} \end{cases} \quad (2.20)$$

where  $q_n \in \mathbb{C}$  are a set of fixed predefined basis poles. The generalized model structure thus reads

$$\mathbf{H}(s) = \frac{\mathbf{N}(s)}{D(s)} = \frac{\mathbf{R}_0 + \sum_{n=1}^{\bar{n}} \frac{\mathbf{R}_n}{s-q_n}}{b_0 + \sum_{n=1}^{\bar{n}} \frac{b_n}{s-q_n}} \quad (2.21)$$

In the following, we will refer to the model functional form (2.21) as *rational barycentric form*, although this term is commonly used only when the constants  $\mathbf{R}_0$  and  $b_0$  are not present.

The first requirement is trivially satisfied, since the ratio of rational functions is still a rational function. With regard to the second constraint, we highlight that a formal conditioning analysis is complex and actually depends on the input data samples. However, we report a significant numerical illustration in Figure 2.1 where we compare the condition numbers  $\kappa_{\Phi}$  of the least-squares matrices  $\Phi$  associated with the polynomial basis, defined in (2.14), and with the partial fraction basis, further defined in (2.28); the condition number  $\kappa_{\mathbf{X}}$  of a generic matrix  $\mathbf{X}$  is defined as

$$\kappa_{\mathbf{X}} := \frac{\bar{\sigma}_{\mathbf{X}}}{\underline{\sigma}_{\mathbf{X}}} \quad (2.22)$$

being  $\bar{\sigma}_{\mathbf{X}}$  and  $\underline{\sigma}_{\mathbf{X}}$  the leading and the least singular values of matrix  $\mathbf{X}$ , respectively. It measures the robustness of its numerical inversion: the larger the condition number, the more sensitive is the inversion to numerical perturbations and round-off errors. The improved conditioning introduced by the partial fraction basis is immediately evident comparing the orders of magnitudes. Obviously, the conditioning properties in the partial fraction basis strongly depend on the location of the fixed poles  $q_n$ , that must be distinct and sufficiently separated.

**Remark 1.** *With the partial fractions defined in (2.20) we can not express poles with multiplicity higher than one. However, as pointed out in [33], for any*

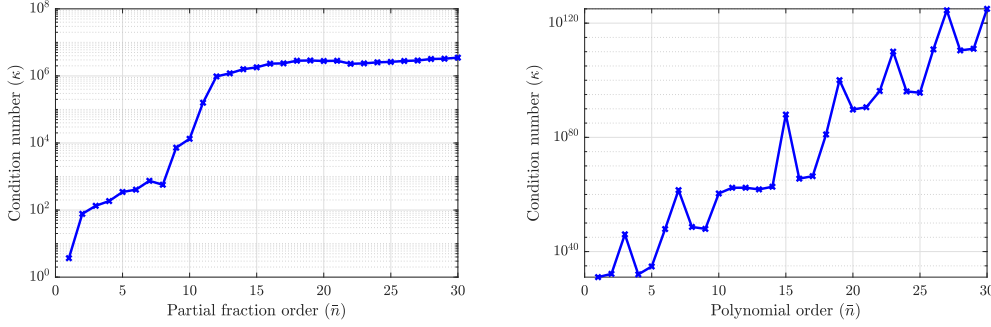


Fig. 2.1 Comparing the conditioning of the SK optimization problems associated with partial fraction basis (left panel) and polynomial basis (right panel)

*polynomial with coincident roots we can find another arbitrarily close polynomial with distinct roots. Therefore, the partial fractions (2.20) are sufficient for our approximation purposes.*

Let us now go through the details of the GSK algorithm. First, let us define the GSK fitting residual  $\mathbf{E}_{k,\nu}^{GSK}$ , defined as in (2.17) with modified  $\mathbf{N}_\nu(s_k)$ ,  $D_\nu(s_k)$ , that now make reference to the more general model structure (2.21). In addition, let us collect the (unknown) model coefficient in vector  $\mathbf{c}$ , as

$$\mathbf{c} = [\mathbf{r}_{(1,1)}^\top, \dots, \mathbf{r}_{(P,P)}^\top, \mathbf{b}^\top]^\top \quad (2.23)$$

where  $\mathbf{r}_{i,j}$  collects the  $\bar{n} + 1$  expansion coefficients of  $\mathbf{R}_n$ , associated with port  $(i, j)$ , and  $\mathbf{b} = [b_0, \dots, b_{\bar{n}}]^\top$ .

At each iteration  $\nu$ , an estimate of the model coefficients  $\mathbf{c}^\nu$  is obtained by minimizing the error  $\mathbf{E}_{k,\nu}^{GSK}$ ,  $k = 1, \dots, K$  in least-squares sense. To this end, we consider each input/output port combination separately, and we set up the following least-squares optimization problem

$$\mathbf{c}^\nu = \arg \min_{\mathbf{c}} \|\Psi^\nu \mathbf{c}\| \quad (2.24)$$

where the regressor matrix is defined as

$$\Psi^\nu = \begin{pmatrix} \Gamma^\nu & & \Xi_{(1,1)}^\nu \\ & \ddots & \vdots \\ & & \Gamma^\nu & \Xi_{(P,P)}^\nu \end{pmatrix} \in \mathbb{C}^{P^2 K \times (P^2+1)(\bar{n}+1)}, \quad (2.25)$$



and

$$\mathbf{\Gamma}^\nu = \mathbf{W}^{\nu-1} \mathbf{\Phi} \in \mathbb{C}^{K \times (\bar{n}+1)}, \quad (2.26)$$

$$\mathbf{\Xi}_{(i,j)}^\nu = -\mathbf{W}^{\nu-1} \check{\mathbf{H}}_{(i,j)} \mathbf{\Phi} \in \mathbb{C}^{K \times (\bar{n}+1)}. \quad (2.27)$$

The matrix  $\mathbf{\Phi}$  contains the evaluated partial fraction bases, as

$$\mathbf{\Phi} = \begin{pmatrix} 1 & \frac{1}{s_1 - q_1} & \cdots & \frac{1}{s_1 - q_{\bar{n}}} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ 1 & \frac{1}{s_K - q_1} & \cdots & \frac{1}{s_K - q_{\bar{n}}} \end{pmatrix} \quad (2.28)$$

and

$$\mathbf{W}^{\nu-1} = \text{diag}\{D_{\nu-1}(s_1), \dots, D_{\nu-1}(s_K)\} \quad (2.29)$$

$$\check{\mathbf{H}}_{(i,j)} = \text{diag}\{\check{H}_{(i,j)}(s_1), \dots, \check{H}_{(i,j)}(s_K)\} \quad (2.30)$$

At the first GSK iteration, the denominator is initialized as  $D_0(s_k) = 1$  (i.e.,  $b_n = 0$  for  $n \geq 1$ ) and, based on (2.18), the iterative scheme is stopped when the denominator estimates  $D_\nu(s_k)$  converge.

**Remark 2.** *It should be pointed out that the solution vector  $\mathbf{c}^\nu$  is unique up to a normalization constant applied to both the model numerator and denominator. This observation may also arise noting that the GSK least-squares (2.24) is homogeneous and admits the vanishing trivial solution. This problem can be circumvented by suitably constraining the optimization problem at each iteration (e.g., by setting  $\|\mathbf{c}^\nu\| = 1$ ). Alternatively, one coefficient can be fixed to a given prescribed constant, e.g.,  $b_0 = 1$ . A thorough discussion on this topic can be found in [34].*

**Remark 3.** *We remark that there is no guarantee that the SK and GSK schemes converge to some fixed equilibrium solution. However, non-converging behaviors (oscillations between two or more fixed points) arise especially when the data are corrupted by large noise components, a scenario that can be considered to be very rare in most practical applications [35].*

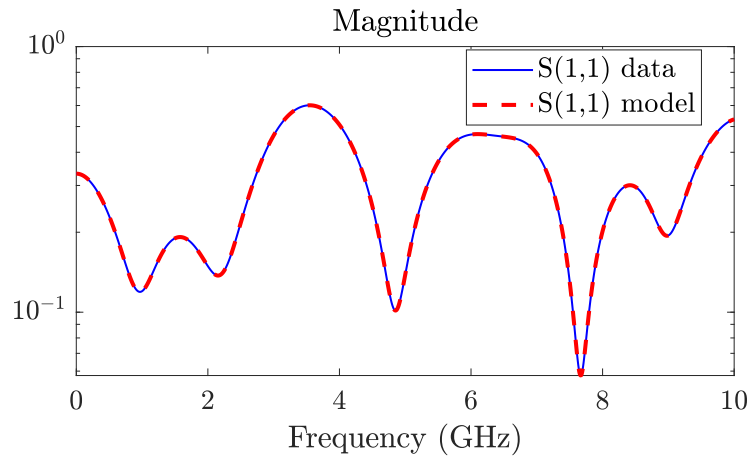


Fig. 2.2 Comparing the GSK rational model responses (red dashed lines) with the available raw data (blue solid line) for the selected transmission line example.

### Numerical example

In order to illustrate the GSK modeling performance, we report a relevant numerical example. We consider a transmission line network, composed of four cascaded lossy line segments, with three internal loaded stubs. More details on this structure can be found in Section 4.5.3, where it is fully described in its parametric form. For this numerical test, we fix the lines and stubs lengths to 7 mm and 9 mm, respectively, while the loads' reflection coefficients are set to their nominal values  $\Gamma = 0.5$ .

The GSK scheme is used to construct a rational model for the input/output scattering parameters in the band [1 Hz, 10 GHz], assuming a number of poles  $\bar{n} = 20$ . The worst-case relative error is in the order of  $10^{-4}$ , as confirmed by Figure 2.2, that compares the response  $S_{1,1}$  of the data (blue solid line) and the model (red-dashed line).

### Discussion

Despite the lack of theoretical guarantee of convergence, over the years the GSK scheme based on partial fraction basis has proven to be a very effective tool for rational approximation. However, the GSK algorithm provides accurate results only assuming that the set of basis poles  $q_n$  is somehow close to the system poles, which are often unknown. Hence, the need of additional information

regarding the device under modeling prevents the GSK scheme to be interpreted as a fully black-box modeling approach.

### 2.2.6 Vector fitting

The problem of finding a “good” set of basis poles remained unsolved for several years until, in 1999, B. Gustavsen and A. Semlyen first presented the so-called *Vector Fitting* (VF) algorithm in [36]. The idea behind the vector fitting scheme is simple: starting with a set of arbitrarily chosen *initial* poles  $\{q_n\}_{n=1,\dots,\bar{n}}$ , the VF algorithm iteratively updates their estimates until they converge to the set of poles  $\{p_n\}_{n=1,\dots,\bar{n}}$  that best fits the available data. We remark that, in case the data come from a lumped parameters circuit (or from an electrically distributed structure for which the lumped approximation holds), the set  $\{p_n\}_{n=1,\dots,\bar{n}}$  coincide with its dominant poles.

Let us now briefly describe the algorithm. As in the previous SK and GSK settings, we assume to be given with sampled frequency responses  $\check{\mathbf{H}}(s_k)$ , evaluated over a prescribed band.

In the first initialization phase of the VF algorithm, we must select a set of *initial* poles  $\{q_n\}$ . Although this choice may be completely arbitrary, a thoughtful selection of these poles may dramatically improve the numerical conditioning. A common practice is to choose a set of weakly damped complex conjugate poles with imaginary part linearly spaced over the frequency band of interest [19]. As this set of poles is iteratively updated, we denote the set of poles at iteration  $\nu$  as  $\{q_n\}^\nu = \{q_1^\nu, \dots, q_{\bar{n}}^\nu\}$ . The associated partial fractions are therefore identified as  $\varphi_n^\nu(s)$ .

The basic vector fitting iteration unfolds in 3 main steps. First, assuming to be at the  $\nu$ -th iteration, we use the GSK barycentric model form defined in (2.21) to set up the approximation

$$\mathbf{H}^\nu(s_k) = \frac{\mathbf{N}^\nu(s)}{D^\nu(s)} = \frac{\mathbf{R}_0^\nu + \sum_{n=1}^{\bar{n}} \frac{\mathbf{R}_n^\nu}{s_k - q_n^{\nu-1}}}{b_0^\nu + \sum_{n=1}^{\bar{n}} \frac{b_n^\nu}{s_k - q_n^{\nu-1}}} \approx \check{\mathbf{H}}(s_k) \quad (2.31)$$

By assumption, the current basis poles  $\{q_n^{\nu-1}\}$  are known from the previous iteration; thus, the unknowns  $\{\mathbf{R}_n^\nu\}$ ,  $\{b_n^\nu\}$  can be readily computed solving

$$\min \left\| \mathbf{R}_0^\nu + \sum_{n=1}^{\bar{n}} \frac{\mathbf{R}_n^\nu}{s_k - q_n^{\nu-1}} - \check{\mathbf{H}}(s_k) \left( b_0^\nu + \sum_{n=1}^{\bar{n}} \frac{b_n^\nu}{s_k - q_n^{\nu-1}} \right) \right\|_F \quad (2.32)$$

in least-squares sense (for additional technical details, see [19, 34, 37]).

Once both the coefficients  $\mathbf{R}_n^\nu$ ,  $b_n^\nu$  and the fixed poles  $\{q_n^{\nu-1}\}$  are available, we can transform the model (2.31) into a classical pole-zeros representation. The application of this transformation enables to highlight the estimates  $\{p_n^\nu\}$  of the real system's poles, that can be easily computed resorting to a state-space realization of the denominator  $D^\nu(s)$  that is constructed as

$$D^\nu(s) \rightarrow \left( \begin{array}{c|c} \mathbf{A}^\nu & \mathbf{1} \\ \mathbf{b}^\nu & b_0^\nu \end{array} \right) \quad (2.33)$$

where  $\mathbf{A}^\nu = \text{diag}\{q_1^{\nu-1}, \dots, q_{\bar{n}}^{\nu-1}\}$  and  $\mathbf{b}^\nu = (b_1^\nu, \dots, b_{\bar{n}}^\nu)$ . The poles  $\{p_n^\nu\}$  are the *zeros* of the denominator, that can be thus computed as the poles of its inverse  $D^\nu(s)^{-1}$ , that has the following state-space representation

$$\left( D^\nu(s) \right)^{-1} \rightarrow \left( \begin{array}{c|c} \mathbf{A}^\nu - \mathbf{1}(b_0^\nu)^{-1}(\mathbf{b}^\nu)^\top & \mathbf{1}(b_0^\nu)^{-1} \\ (b_0^\nu)^{-1}\mathbf{b}^\top & (b_0^\nu)^{-1} \end{array} \right) \quad (2.34)$$

Hence, the poles  $\{p_n^\nu\}$  are evaluated as

$$\{p_n^\nu\} = \text{eig}\left(\mathbf{A}^\nu - \mathbf{1}(b_0^\nu)^{-1}(\mathbf{b}^\nu)^\top\right) \quad (2.35)$$

where the operator  $\text{eig}(\cdot)$  compute the eigenvalues of its matrix argument.

The last *poles relocation* step is what distinguishes the VF from the previous methods. At this stage, we update (relocate) the set of basis poles to be used at the next iteration  $\nu + 1$  as  $\{q_n^\nu\} = \{p_n^\nu\}$ . This procedure is repeated until convergence, that is reached when the pole estimates stabilize, i.e.,  $\{q_n^\nu\} = \{q_n^{\nu-1}\}$ . Convergence is usually achieved in few iterations, especially in case the data are not corrupted by noise.

We highlight that the new set of poles  $\{q_n^\nu\}$  is not guaranteed to have a negative real part, possibly leading the current model  $\mathbf{H}(s)$  to be unstable. This issue is easily circumvented with the so-called *pole flipping* method: if some

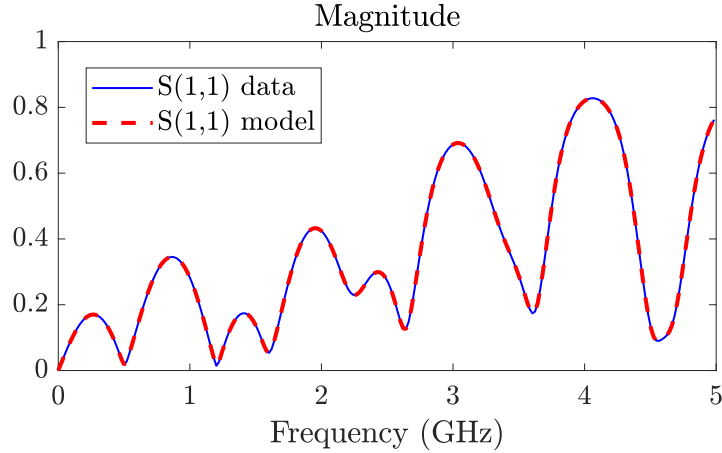


Fig. 2.3 Modeling performances of the vector fitting algorithm on the high-speed link example. The data and model response are depicted with blue-solid line and red-dashed line, respectively.

poles  $q_n'$  have a positive real part, we set  $\Re\{q_n'\} = -\Re\{q_n\}$ . This procedure enforces stability without affecting the model accuracy.

**Remark 4.** *As the GSK, also the VF scheme is not guaranteed to converge. Actually, there are counterexamples that trigger non-converging behaviors [38]. However, these examples are artificial and specifically created to destabilize the algorithm; in general, the VF algorithm converges (even fairly quickly) in the vast majority of practical applications.*

### Numerical example

We demonstrate the modeling performance of the vector fitting algorithm on a non-parametric version of the high-speed link described in Section 4.5.2, consisting of an interconnection between two multi-layer PCBs. In this example, we consider the case where the pad and anti-pad radii are fixed to  $100 \mu\text{m}$  and  $400 \mu\text{m}$ , respectively.

The VF algorithm is used to obtain a rational model of the scattering parameters, in the band  $[0 \text{ Hz}, 5 \text{ GHz}]$ . The modeling procedure has been set up with  $\bar{n} = 24$  poles, and returned a stable model whose accuracy is in the order of  $10^{-4}$ . Figure 2.3 illustrate the modeling accuracy for the  $S_{1,1}$  scattering matrix element. In addition, Figure 2.4 illustrates the convergence of the vector

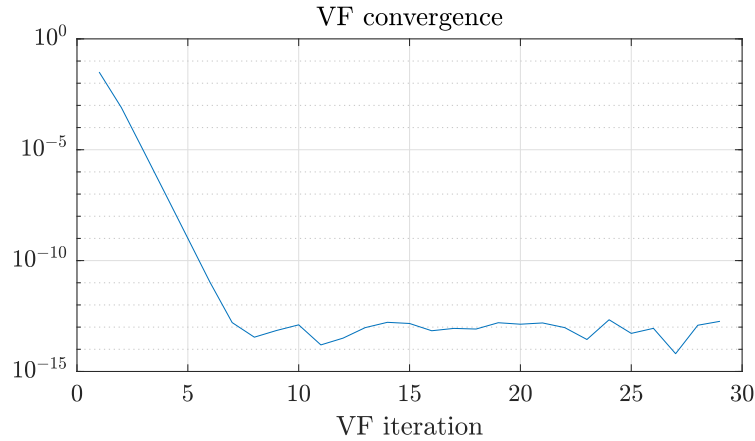


Fig. 2.4 Convergence of the vector fitting algorithm illustrated by measuring the Hausdorff distance between the set of poles  $\{p_n\}$ , evaluated at subsequent iterations.

fitting by measuring the (normalized) distance (in Hausdorff sense [39]) between the set of poles  $\{p_n^\nu\}$  corresponding to two subsequent iterations  $\nu - 1$  and  $\nu$ . Starting from the set of initial poles  $\{p_n^0\}$ , the subsequent estimates converge to an equilibrium solution (the “noise” appearing after the tenth iteration is due to off-band poles that slightly oscillate around their positions).

## Discussion

The remarkable characteristics of the vector fitting algorithm justify the vast literature regarding its extensions. Several efforts have been devoted to improve the VF numerical characteristics using orthonormal rational basis functions [10] and to improve the convergence properties in presence of noisy data [40, 41]. To conclude, we highlight that [26–28] proposed a time-domain VF implementation that, starting from a time-domain responses, evaluate the poles of the equivalent frequency-domain representation.

The relevance of rational approximation in many scientific areas motivated other research efforts. In particular, other well-known approaches are the Loewner method and the related AAA algorithm. In this work we will not use these latter schemes, for which a detailed description can be found in the original publications [42] and [43], respectively.

## 2.3 Parameterized macromodeling

During the industrial design and production phases of electronic and electrical devices, not all the details of the structure are completely defined. For instance, geometrical and/or component values may still be undetermined, as they require a final optimization and design centering step. In addition, we know that the component values are affected by a statistical uncertainty, that often comes in the form of a probabilistic distribution. Hence, assuming some appropriate variability bounds, even at finalized design the input/output behavior of the device is not deterministically defined.

It should also be noted that the electronic components behave differently depending on the environmental working conditions (e.g., room temperature, incident EM fields, etc. . . ), that must be carefully taken into consideration in the design phase. In our view, any of the above variables can be considered as a *parameter* that affects the behavior of the system under modeling.

The availability of reduced order models that embed in closed form the dependence upon such parameters turns out to be extremely helpful, as they allow a dramatic simplification of the simulation processes that would otherwise require an enormous effort both in terms of computational resources and manpower.

The macromodeling techniques described in Section 2.2 are not able to include in the model a parametric variability. These models will be therefore denoted as *univariate*, since the only independent variable is the complex frequency.

Our objective is instead to obtain a macromodel  $\mathbf{H}(s, \boldsymbol{\vartheta})$  that embeds in closed form the dependence upon a (possibly large) set of independent parameters, collected in vector  $\boldsymbol{\vartheta} = [\vartheta^1, \dots, \vartheta^\rho]$ , that affect the input/output behavior of the device under modeling. We assume a limited range of validity for the parameters, that must be bounded within a  $\rho$ -dimensional hypercube  $\Theta$  that we define *parameter space*. These *parameterized* models will be denoted as *multivariate*. We remark that these parameters can be of any nature; the only requirement is that the frequency responses depend continuously on their variations.

### 2.3.1 Parameterized macromodeling flows

Any data-driven parametric macromodeling flow requires a preliminary extraction of parametric frequency responses, either obtained via simulation or measurements. Throughout this work, we will denote the generic  $m$ -th parametric response as

$$\check{\mathbf{H}}_m = \check{\mathbf{H}}(s_k, \boldsymbol{\vartheta}_m), \quad k = 1, \dots, K \quad (2.36)$$

composed of  $K$  frequency samples. All the  $\bar{m}$  available parameterized frequency responses are then collected in set  $\mathcal{D}$ , defined as

$$\mathcal{D} := \{\check{\mathbf{H}}_m : m = 1, \dots, \bar{m}\} \quad (2.37)$$

We assume this parametric dataset to be defined a-priori, hence we do not consider the availability of on-demand solver calls to interactively or adaptively gather parameterized samples. In literature, some efforts have been devoted to investigate this latter possibility, see [44–46].

Based on these available parametric data, we build a model that must reproduce the broadband frequency response over all the feasible parameter instances; it must retain a rational frequency dependence, while the parametric variability can be embedded in different ways. In the past years, several approaches have been presented to synthesize such models, each coming with its pros and cons. For instance, some methods are capable of guaranteeing stability and passivity by construction, at the cost of generating non-compact models that can not be directly implemented as black-box surrogates in commercial solvers (e.g., SPICE). Others, conversely, are capable of generating compact parametric models (thus ready to use in circuit solvers) that, however, require advanced stability and passivity enforcement techniques.

In the following, we will briefly review the most relevant approaches.

### 2.3.2 Interpolation-based methods

To begin, we introduce what can be considered one of the most intuitive parameterized macromodeling approaches, hereinafter referred to as *interpolation based*.



This methodology has been introduced by Ferranti et al. in [47]; successive and more refined strategies have been later presented in [48, 8, 7, 9]. All these methods assume the parameterized macromodel to be constructed as a linear combination of some reference non-parameterized (root) macromodels.

The first macromodeling step, common to all these strategies, consists in the discretization of the parameter space in  $\rho$ -dimensional mutually disjoint convex volumes  $\mathcal{V}_k$ , each identified by  $N_v$  vertices. For the generic cell  $\mathcal{V}_k$ , the vertices locations are identified as  $\boldsymbol{\vartheta}_k^i$ , with  $i = 1, \dots, N_v$ .

Once the parameter space has been suitably discretized, the next step is to synthesize the so-called *root macromodels*. Basically, the root macromodels  $\hat{\mathbf{R}}(s, \boldsymbol{\vartheta}_k^i)$  are reference non-parameterized macromodels (obtained, for instance, through vector fitting) constructed in correspondence to the cell vertices  $\boldsymbol{\vartheta}_k^i$ , as

$$\hat{\mathbf{R}}(s, \boldsymbol{\vartheta}_k^i) = \hat{\mathbf{R}}_0(\boldsymbol{\vartheta}_k^i) + \sum_{n=1}^{\bar{n}} \frac{\hat{\mathbf{R}}_n(\boldsymbol{\vartheta}_k^i)}{s - q_n^k} \quad (2.38)$$

The root macromodels are enforced to be stable (through pole-flipping) and passive (usually with a-posteriori passivity enforcement procedures, e.g., [18]).

We remark that, in general, the set of basis poles  $\{q_n^k\}$  is different for each considered vertex.

The multi-variate macromodel is obtained as a linear combination (further denoted as *interpolation*) of different features of the root macromodels (e.g., transfer matrices, state-space matrices, etc...), which we identify with a common placeholder  $\Upsilon(s, \boldsymbol{\vartheta})$ . More in details, for each elementary cell  $\mathcal{V}_k$  with vertices  $\boldsymbol{\vartheta}_i^k$ , we gather a set of precomputed instances of the model features  $\hat{\Upsilon}(s, \boldsymbol{\vartheta}_i^k)$ ,  $i = 1, \dots, N_v$ . In correspondence to a generic parameter combination  $\boldsymbol{\vartheta}$  located inside the cell  $\mathcal{V}_k$ , the (local) parameterized model feature is defined as

$$\Upsilon(s, \boldsymbol{\vartheta}) = \sum_{i=1}^{N_v} \hat{\Upsilon}(s, \boldsymbol{\vartheta}_i^k) I_i^k(\boldsymbol{\vartheta}) \quad (2.39)$$

where  $I_i^k(\boldsymbol{\vartheta})$  are multi-linear interpolation kernel, locally defined in the cell  $\mathcal{V}_k$ .

The most compelling feature of the interpolation approaches is the possibility of guaranteeing (locally inside  $\mathcal{V}_k$ ) stability and passivity by construction. Depending on the considered strategy, this is enabled by carefully selecting

kernels  $I_i^k(\boldsymbol{\vartheta})$  that satisfy the following conditions

$$I_i^k(\boldsymbol{\vartheta}) \geq 0, \quad (2.40)$$

$$I_i^k(\boldsymbol{\vartheta}^j) = \delta_{i,j}, \quad (2.41)$$

$$\sum_{i=1}^{N_v} I_i^k(\boldsymbol{\vartheta}) = 1. \quad (2.42)$$

Among the possible choices, all the above-cited works define  $I_i^k(\boldsymbol{\vartheta})$  as piece-wise functions, suitably defined in order to satisfy (2.40).

Depending on the target of the interpolation procedure, we divide the above in two categories [7–9, 47, 48]:

- *Input/Output methods:*  
perform the interpolation of the (possibly modified) transfer matrices of the root macromodels; this category includes the *residue interpolation* and *shifted/scaled interpolation*.
- *State-space methods:*  
perform the interpolation of the state space matrices; this category includes the *direct state-space interpolation* and *Sylvester interpolation*.

## Residues interpolation

In the *input/output* category, the first approach has been presented in [47] and relies on the interpolation of just the model residues. In fact, it is not advisable to directly parameterize the model poles, whose parametric trajectories may undergo non-smooth transitions (e.g., bifurcations) and/or may exhibit high sensitivity to parameter variations.

Based on the multi-variate interpolation method (2.39), we define  $\Upsilon(s, \boldsymbol{\vartheta}) = \mathbf{R}(s, \boldsymbol{\vartheta})$  and  $\hat{\Upsilon}(s, \boldsymbol{\vartheta}_i^k) = \hat{\mathbf{R}}(s, \boldsymbol{\vartheta}_i^k)$ . Thus, for a generic parameter instance  $\boldsymbol{\vartheta}$  belonging to the elementary cell  $\mathcal{V}_k$ , the (local) multi-variate parameterized model assumes the form

$$\mathbf{R}(s, \boldsymbol{\vartheta}) = \sum_{i=1}^{N_v} \hat{\mathbf{R}}(s, \boldsymbol{\vartheta}_i^k) I_i^k(\boldsymbol{\vartheta}) \quad (2.43)$$

Along with the simplicity of this technique, there are several drawbacks:

1. The presence of non-parameterized poles is a strong limitation since, in general, even small parameter variations may induce large modifications to the system dynamics (e.g., parameterized resonances). The assumption of fixed poles strongly confines the applicability of this method to a restricted set of applications.
2. The resulting parameterized model  $\mathbf{R}(s, \boldsymbol{\vartheta})$  is, in general, more complex than strictly required. In fact, the model (2.43) retains *all* the poles of the individual root models  $\hat{\mathbf{R}}(s, \boldsymbol{\vartheta}_i^k)$ . In the worst-case scenario, in which all the models  $\hat{\mathbf{R}}(s, \boldsymbol{\vartheta}_i^k)$ ,  $i = 1, \dots, N_v$  have a different set of poles  $\{q_n^k\}_{n=0}^{\bar{n}}$ , the dynamic order of the resulting parameterized model would be  $N_v(\bar{n} + 1)$  which is, of course, too large.

### Indirect poles interpolation

This second *input/output* approach [7] seeks to parameterize, in addition to the residues, also the model poles, by introducing an intermediate indirect parameterization step. Starting from the initial set of root macromodels  $\mathbf{R}(s, \boldsymbol{\vartheta}_i^k)$ , for each vertex  $\boldsymbol{\vartheta}_i^k$  a set of scaling and shifting coefficients  $\alpha_{1,i}(\boldsymbol{\vartheta}_j^k)$  and  $\alpha_{2,i}(\boldsymbol{\vartheta}_j^k)$ ,  $j = 1, \dots, N_v$  is computed, so that the shifted and scaled model

$$\tilde{\mathbf{R}}(s, \boldsymbol{\vartheta}_i^k) = \alpha_{1,i}(\boldsymbol{\vartheta}_j^k) \mathbf{R}(\alpha_{2,i}(\boldsymbol{\vartheta}_j^k) s, \boldsymbol{\vartheta}_i^k) \quad (2.44)$$

is as close as possible to the other root models  $\mathbf{R}(s, \boldsymbol{\vartheta}_j^k)$ , for each  $i, j = 1, \dots, N_v$ . See [7] for more details.

Then, the estimates of coefficients  $\alpha_{1,i}(\boldsymbol{\vartheta})$ ,  $\alpha_{2,i}(\boldsymbol{\vartheta})$  evaluated at a generic parameter combination  $\boldsymbol{\vartheta}$  are obtained through (2.39) by interpolation, as

$$\alpha_{1,i}(\boldsymbol{\vartheta}) = \sum_{j=1}^{N_v} \alpha_{1,i}(\boldsymbol{\vartheta}_j^k) I_j^k(\boldsymbol{\vartheta}) \quad (2.45)$$

and similarly for  $\alpha_{2,i}(\boldsymbol{\vartheta})$ .

Finally, the multivariate macromodel evaluated at  $\boldsymbol{\vartheta}$  is obtained interpolating at an input-output level, as

$$\mathbf{R}(s, \boldsymbol{\vartheta}) = \sum_{i=1}^{N_v} \alpha_{1,i}(\boldsymbol{\vartheta}) \mathbf{R}(\alpha_{2,i}(\boldsymbol{\vartheta}) s, \boldsymbol{\vartheta}_i^k) I_i^k(\boldsymbol{\vartheta}) \quad (2.46)$$

Similarly to the *residue interpolation* approach, combining (in this case shifted and scaled) transfer matrices lead, in general, the resulting multi-variate model  $\mathbf{R}(s, \boldsymbol{\vartheta})$  to be excessively complex.

In contrast, instead, this second strategy allows for a pole parameterization induced by the shifting coefficients. However, being  $\alpha_{2,i}(\boldsymbol{\vartheta})$  only multiplicative constants, they do not allow for a complete control over the poles parameterization and act only as scaling constants over all the poles.

### State-space interpolation

As a result of the input/output interpolation strategy, both the above approaches produce models with higher complexity than necessary. This problem has originally been tackled in [8] by resorting to a state-space description of the root macromodels. Hence, denoting as usual the root model located at the vertex  $\boldsymbol{\vartheta}_i^k$  as  $\mathbf{R}(s, \boldsymbol{\vartheta}_i^k)$ , let us define the associated state-space realization matrices as  $\{\mathbf{A}_i^k, \mathbf{B}_i^k, \mathbf{C}_i^k, \mathbf{D}_i^k\}$ .

Let us assume that *all* the root macromodels  $\mathbf{R}(s, \boldsymbol{\vartheta}_i^k)$  are guaranteed passive, therefore there must necessarily exist a symmetric positive definite *Lyapunov* matrix  $\mathbf{P}_i^k$  that satisfies the *Positive Real Lemma* or the *Bounded Real Lemma* (see [8, 49] and Section 3.2). This matrix is computed for all the available root models (for advanced details about its practical evaluation, see [8]) and used to project the original state-space representation onto an equivalent non-impulsive descriptor form, as

$$\left( \begin{array}{c|c} [\mathbf{P}_i^k \mathbf{A}_i^k, \mathbf{P}_i^k] & \mathbf{P}_i^k \mathbf{B}_i^k \\ \hline \mathbf{C}_i^k & \mathbf{0} \end{array} \right) \quad (2.47)$$

corresponding to the local root macromodel

$$\mathbf{R}(s, \boldsymbol{\vartheta}_i^k) = \mathbf{C}_i^k \left( s\mathbf{P}_i^k - \mathbf{P}_i^k \mathbf{A}_i^k \right)^{-1} \mathbf{P}_i^k \mathbf{B}_i^k \quad (2.48)$$

It should be noted that the realizations (2.47) are not unique, since constructed upon Lyapunov matrices that are computed independently for each vertex.

Then, exploiting the *Lur'e* equations [50], the descriptor form (2.48) can be re-written introducing  $\mathbf{G}_i^k = -\mathbf{P}_i^k \mathbf{A}_i^k$ , as

$$\mathbf{R}(s, \boldsymbol{\vartheta}_i^k) = \mathbf{C}_i^k (s\mathbf{P}_i^k + \mathbf{G}_i^k)^{-1} (\mathbf{C}_i^k)^\top \quad (2.49)$$

where the properties  $\mathbf{P}_i^k \mathbf{B}_i^k = (\mathbf{C}_i^k)^\top$ ,  $\mathbf{P}_i^k = (\mathbf{P}_i^k)^\top$  and the positive definiteness of  $\mathbf{G}_i^k + (\mathbf{G}_i^k)^\top$  guarantee the passivity.

Finally, the parameterized macromodel is obtained by suitably gluing the *root* descriptor realization matrices  $\mathbf{C}_i^k$ ,  $\mathbf{P}_i^k$ ,  $\mathbf{G}_i^k$ ,  $\mathbf{P}_i^k \mathbf{B}_i^k$ . If we assume a generic parameter instance  $\boldsymbol{\vartheta}$  belonging to the cell  $\mathcal{V}_k$ , we achieve a (local) parameterization of the above matrices through the multi-linear interpolation (2.39), as

$$\mathbf{C}^k(\boldsymbol{\vartheta}) = \sum_{i=1}^{N_v} \mathbf{C}_i^k I_i^k(\boldsymbol{\vartheta}) \quad (2.50)$$

and similarly for  $\mathbf{P}^k(\boldsymbol{\vartheta})$ ,  $\mathbf{G}^k(\boldsymbol{\vartheta})$ ,  $(\mathbf{P}^k \mathbf{B}^k)(\boldsymbol{\vartheta})$ . Based on these interpolated realization matrices, we can construct the (local) parameterized macromodel  $\mathbf{R}(s, \boldsymbol{\vartheta})$ .

The use of positive definite interpolation kernels  $I_i(\boldsymbol{\vartheta})$  guarantees that the final parameterized model  $\mathbf{R}(s, \boldsymbol{\vartheta})$  is uniformly passive within the considered elementary cell  $\mathcal{V}_k$ .

A direct interpolation in the state-space has two main advantages:

1. the final parameterized model does not suffer from an increased complexity, as in the case of input/output methods, and
2. the parameterization of the poles is more flexible, boosting the modeling capabilities.

However, as pointed out in [7, 9], the direct interpolation of realization matrices may give rise to some problems, such as the lack of smoothness of the parameter-dependent interpolated state-space matrices.

### Sylvester's realization interpolation

To circumvent this problem, it would be required to project the realization matrices onto a common parameter-invariant subspace; in fact, performing interpolations in this space would guarantee the smoothness of the resulting parameterized matrices.

This idea has been developed in [9], where the parameter-invariant *Sylvester's* realization is introduced, defined for the root macromodel  $\mathbf{R}(s, \boldsymbol{\vartheta}_i^k)$  as

$$\left( \begin{array}{c|c} \tilde{\mathbf{A}} - \tilde{\mathbf{B}}_i^k \mathbf{F} & \tilde{\mathbf{B}}_i^k \\ \hline \tilde{\mathbf{C}}_i^k & \tilde{\mathbf{D}}_i^k \end{array} \right) \quad (2.51)$$

The terms  $\tilde{\mathbf{A}}$ ,  $\mathbf{F}$  are suitably defined pivot matrices, that we assume to be known at this stage. Initially, the root models  $\mathbf{R}(s, \boldsymbol{\vartheta}_i^k)$  are characterized by a canonical realization, associated with matrices  $\{\mathbf{A}_i^k, \mathbf{B}_i^k, \mathbf{C}_i^k, \mathbf{D}_i^k\}$ . Then, the solutions  $\mathbf{X}_i^k$  of the following *Sylvester's* equation

$$\mathbf{A}_i^k \mathbf{X}_i^k - \mathbf{X}_i^k \tilde{\mathbf{A}} + \mathbf{B}_i^k \mathbf{F} = \mathbf{0} \quad (2.52)$$

are used as similarity transformations to project the original realization onto the *Sylvester's* base (2.51), as

$$\tilde{\mathbf{A}} - \tilde{\mathbf{B}}\mathbf{F} = (\mathbf{X}_i^k)^{-1} \mathbf{A}_i^k \mathbf{X}_i^k \quad (2.53)$$

$$\tilde{\mathbf{B}}_i^k = (\mathbf{X}_i^k)^{-1} \mathbf{B}_i^k \quad (2.54)$$

$$\tilde{\mathbf{C}}_i^k = \mathbf{C}_i^k \mathbf{X}_i^k \quad (2.55)$$

$$\tilde{\mathbf{D}}_i^k = \mathbf{D}_i^k \quad (2.56)$$

In [9] it has been pointed out that if the following conditions hold

1. the pair  $(\tilde{\mathbf{A}}, \mathbf{F})$  is observable,
2. the intersection of the eigenspectra of  $\mathbf{A}_i^k$  and  $\tilde{\mathbf{A}}$  is empty,
3. the pair  $(\mathbf{A}_i^k, \mathbf{B}_i^k)$  is observable

the Sylvester's equation (2.52) has a unique and non-singular solution. Hence, realization matrices that are projected onto a Sylvester's form share the same

basis and the safely interpolated without the risk of non-smooth results. Some practical rules to select appropriate pivot matrices fulfilling the above constraints are reported in [9].

For each root model we thus compute the matrix  $\mathbf{X}_i^k$ , used to project the standard realization  $\{\mathbf{A}_i^k, \mathbf{B}_i^k, \mathbf{C}_i^k, \mathbf{D}_i^k\}$  onto the common Sylvester's form. Then, following the same procedure detailed in the *Direct State-Space Interpolation* method, and assuming the parameter instance  $\boldsymbol{\vartheta}$  to be located inside the elementary cell  $\mathcal{V}_k$ , we set up a local parameterization of the Sylvester's realization matrices, as

$$\tilde{\mathbf{P}}_k(\boldsymbol{\vartheta}) = \sum_{i=1}^{N_v} \tilde{\mathbf{P}}_i^k I_i^k(\boldsymbol{\vartheta}) \quad (2.57)$$

and similarly for  $\tilde{\mathbf{D}}^k(\boldsymbol{\vartheta})$ ,  $\tilde{\mathbf{C}}^k(\boldsymbol{\vartheta})$  and  $(\tilde{\mathbf{P}}^k \tilde{\mathbf{B}}^k)(\boldsymbol{\vartheta})$ .

The auxiliary root matrices  $\tilde{\mathbf{P}}_i^k$  are *Lyapunov* matrices associated with Positive Real and Bounded Real Lemmas, suitably modified to match the new Sylvester's realization.

The resulting parameterized model  $\mathbf{R}(s, \boldsymbol{\vartheta})$ , defined as

$$\mathbf{R}(s, \boldsymbol{\vartheta}) = \tilde{\mathbf{C}}^k(\boldsymbol{\vartheta}) \left( s \tilde{\mathbf{P}}^k(\boldsymbol{\vartheta}) - \tilde{\mathbf{P}}^k(\boldsymbol{\vartheta}) \tilde{\mathbf{A}}^k(\boldsymbol{\vartheta}) \right)^{-1} \tilde{\mathbf{P}}^k(\boldsymbol{\vartheta}) \tilde{\mathbf{B}}^k(\boldsymbol{\vartheta}) + \tilde{\mathbf{D}}^k(\boldsymbol{\vartheta}) \quad (2.58)$$

As in the previous cases, if the interpolation kernels  $I_i(\boldsymbol{\vartheta})$  are positive definite, the multi-variate model  $\mathbf{R}(s, \boldsymbol{\vartheta})$  is guaranteed to be locally (within the cell  $\mathcal{V}_k$ ) uniformly passive.

## Discussion

All the above interpolation methods enable to synthesize accurate parameterized macromodels that are ensured to be passive by construction over the whole parameter space and do not require a-posteriori passivity enforcement runs.

However, there are some non-negligible drawbacks. First, it is not possible to obtain a closed functional model form. This stems from the *local* nature of the parameterized model, that limits its validity to the considered elementary cell. Therefore, such a model is fully defined only in case the entire set of root macromodels is provided. This strongly undermines the applicability of these

models as compact surrogates of circuit blocks to be used in commercial solvers (e.g., SPICE).

In addition, it should be noted that the accuracy and the complexity of the final model heavily depends on the number and the placement of the root macromodels. This problem has been tackled in [51] and [52], where adaptive algorithms based on the iterative (sub-gridding) partitioning of the parameter space in smaller hyper-rectangles is used to determine a sub-optimal set of root macromodels.

These approaches can be applied successfully in case of a reduced number of parameters, but they are limited by the adopted sub-gridding partition rule, whose computational complexity scales badly in case of several independent parameters. Therefore, the number of root macromodels required to fully define the parameterized model may quickly blow up, undermining the applicability and portability of these approaches.

### 2.3.3 Parameterized Sanathanan-Koerner iteration

The last parameterized modeling method we describe is the so-called *Parameterized Sanathanan-Koerner* (PSK) method. This approach has been originally introduced in [10, 11] as a parametric generalization of the GSK iteration, summarized in Section 2.2.5. The PSK method makes use of the same GSK barycentric model form, that we report here for completeness

$$\mathbf{H}(s) = \frac{\sum_{n=0}^{\bar{n}} \mathbf{R}_n \varphi_n(s)}{\sum_{n=0}^{\bar{n}} r_n \varphi_n(s)} = \frac{\mathbf{R}_0 + \sum_{n=1}^{\bar{n}} \frac{\mathbf{R}_n}{s-q_n}}{r_0 + \sum_{n=1}^{\bar{n}} \frac{r_n}{s-q_n}} \quad (2.59)$$

The parametric dependence is embedded in the model under the following assumptions:

1. the model common basis poles  $q_n$  that build the barycentric model form are not parameterized and are not relocated during the model identification procedure;
2. only the numerator and denominator coefficients  $\mathbf{R}_n, r_n$  are in charge of inducing a smooth parameterization on the model responses.



Under these assumptions, the model takes the form

$$\mathbf{H}(s, \boldsymbol{\vartheta}) = \frac{\mathbf{N}(s, \boldsymbol{\vartheta})}{D(s, \boldsymbol{\vartheta})} = \frac{\sum_{n=0}^{\bar{n}} \mathbf{R}_n(\boldsymbol{\vartheta}) \varphi_n(s)}{\sum_{n=0}^{\bar{n}} r_n(\boldsymbol{\vartheta}) \varphi_n(s)} = \frac{\mathbf{R}_0(\boldsymbol{\vartheta}) + \sum_{n=1}^{\bar{n}} \frac{\mathbf{R}_n(\boldsymbol{\vartheta})}{s-q_n}}{r_0(\boldsymbol{\vartheta}) + \sum_{n=1}^{\bar{n}} \frac{r_n(\boldsymbol{\vartheta})}{s-q_n}} \quad (2.60)$$

The parameterized coefficients are assumed to be a (multi)-linear expansion of suitable basis functions  $\xi_{\ell}(\boldsymbol{\vartheta})$ , as

$$\mathbf{R}_n(\boldsymbol{\vartheta}) = \sum_{\ell \in \mathcal{I}_N} \mathbf{R}_{n,\ell} \xi_{\ell}(\boldsymbol{\vartheta}) = \sum_{\ell \in \mathcal{I}_N} \mathbf{R}_{n,\ell} \xi_{\ell_1}(\vartheta^1) \times \cdots \times \xi_{\ell_\rho}(\vartheta^\rho) \quad (2.61a)$$

$$r_n(\boldsymbol{\vartheta}) = \sum_{\ell \in \mathcal{I}_D} r_{n,\ell} \xi_{\ell}(\boldsymbol{\vartheta}) = \sum_{\ell \in \mathcal{I}_D} r_{n,\ell} \xi_{\ell_1}(\vartheta^1) \times \cdots \times \xi_{\ell_\rho}(\vartheta^\rho) \quad (2.61b)$$

where  $\ell = [\ell_1, \dots, \ell_\rho]$  is a multi-index that attains values within the sets  $\mathcal{I}_N$  and  $\mathcal{I}_D$ , that are defined as

$$\mathcal{I}_N := \{\ell_i \in \mathbb{N} : \ell_i \leq \bar{\ell}_i^N\} \quad (2.62a)$$

$$\mathcal{I}_D := \{\ell_i \in \mathbb{N} : \ell_i \leq \bar{\ell}_i^D\} \quad (2.62b)$$

For later use, we define as  $|\mathcal{I}_D|$  and  $|\mathcal{I}_N|$  the cardinality of sets  $\mathcal{I}_N$  and  $\mathcal{I}_D$ , respectively.

**Remark 5.** *The selection of appropriate parameter basis functions  $\xi_{\ell}(\boldsymbol{\vartheta})$  is fundamental to ensure an appropriate parameterization of the model responses. In case no a-priori information are available, it is common to set the basis functions as (first kind) Chebyshev's polynomial, due to their favorable numerical properties. On the contrary, in case the effects of the parameter variation on the data is known, ad-hoc bases should be used (see [53]) to improve accuracy and interpretability.*

Plugging the definitions (2.61) in (2.60) leads to the final PSK model form, that reads

$$\mathbf{H}(s, \boldsymbol{\vartheta}) = \frac{\mathbf{N}(s, \boldsymbol{\vartheta})}{D(s, \boldsymbol{\vartheta})} = \frac{\sum_{n=0}^{\bar{n}} \sum_{\ell \in \mathcal{I}_N} \mathbf{R}_{n,\ell} \xi_{\ell}(\boldsymbol{\vartheta}) \varphi_n(s)}{\sum_{n=0}^{\bar{n}} \sum_{\ell \in \mathcal{I}_D} r_{n,\ell} \xi_{\ell}(\boldsymbol{\vartheta}) \varphi_n(s)} \quad (2.63)$$

It should be highlighted that such a model structure allows for an implicit pole parameterization; this can be easily verified noting that, simplifying the common fixed denominators  $1/(s - q_n)$ , we obtain a standard parameterized pole/residue form,

$$\mathbf{H}(s, \boldsymbol{\vartheta}) = \frac{\sum_{\ell \in \mathcal{I}_N} \sum_{n=0}^{\bar{n}} \mathbf{R}_{n,\ell} \xi_{\ell}(\boldsymbol{\vartheta}) \varphi_n(s)}{\sum_{\ell \in \mathcal{I}_D} \sum_{n=0}^{\bar{n}} r_{n,\ell} \xi_{\ell}(\boldsymbol{\vartheta}) \varphi_n(s)} = \mathbf{P}_0(\boldsymbol{\vartheta}) + \sum_{n=1}^{\bar{n}} \frac{\mathbf{P}_n(\boldsymbol{\vartheta})}{s - p_n(\boldsymbol{\vartheta})} \quad (2.64)$$

where the model poles  $p_n(\boldsymbol{\vartheta})$  are indirectly parameterized as the zeros of the denominator function [53].

We finally remark that, in order to be compliant with the physics of the problem, the model  $\mathbf{H}(s, \boldsymbol{\vartheta})$  must have a real-valued impulse response, which is guaranteed whenever  $\mathbf{H}(s^*, \boldsymbol{\vartheta}) = \mathbf{H}^*(s, \boldsymbol{\vartheta})$ . This constraint is embedded in the model by enforcing

1.  $\mathbf{R}_{n,\ell} \in \mathbb{R}$ , if  $q_n \in \mathbb{R}$ , and
2.  $\mathbf{R}_{n,\ell} = \mathbf{R}_{n+1,\ell}^*$ , if  $q_n = q'_n + q''_n \in \mathbb{C}$  and  $q_{n+1} = q_n^*$

and similarly on the denominator coefficients  $r_{n,\ell}$ .

## Model identification

In order to properly set up model identification, we split the parameterized dataset  $\mathcal{D}$  defined in (2.37) in two mutually disjoint subsets  $\mathcal{T}$  and  $\mathcal{V}$ , each composed of  $m^t$  and  $m^v$  samples, respectively. The elements collected in the *training set*  $\mathcal{T}$  are used to construct the model, while the *validation set*  $\mathcal{V}$  is used to validate the model accuracy. In order to guarantee a uniform accuracy over  $\Theta$ , both the training and validation subsets are selected to homogeneously cover the parameter space.

Based on the above definitions, the numerator and denominator unknowns  $\mathbf{R}_{n,\ell}$ ,  $r_{n,\ell}$  are evaluated enforcing the accuracy on the elements of the training set  $\mathcal{T}$ , as

$$\left\| \frac{\mathbf{N}(s_k, \boldsymbol{\vartheta}_m)}{D(s_k, \boldsymbol{\vartheta}_m)} - \check{\mathbf{H}}(s_k, \boldsymbol{\vartheta}_m) \right\|_F \approx 0 \quad (2.65)$$

This results in a non-convex optimization problem in the decision variables  $\mathbf{R}_{n,\ell}$ ,  $r_{n,\ell}$  that closely matches the univariate GSK rational fitting problem (2.9). The approach devised in [11] thus applies the same relaxation scheme, iteratively solving

$$\left\| \frac{\mathbf{N}^\nu(s_k, \boldsymbol{\vartheta}_m) - D^\nu(s_k, \boldsymbol{\vartheta}_m) \check{\mathbf{H}}(s_k, \boldsymbol{\vartheta}_m)}{D^{\nu-1}(s_k, \boldsymbol{\vartheta}_m)} \right\|_F \approx 0, \quad \nu = 1, 2, \dots \quad (2.66)$$

where the numerator and denominator decision variables appear linearly. The intermediate solutions  $\mathbf{c}^\nu$  of (2.66) are thus computed as

$$\mathbf{c}^\nu = \arg \min_{\mathbf{c}} \|\Psi^\nu \mathbf{c}\| \quad (2.67)$$

with  $\mathbf{c}^\nu = [\mathbf{r}_{(1,1)}^\top, \dots, \mathbf{r}_{(P,P)}^\top, \mathbf{b}^\top]^\top$ ; the vector  $\mathbf{r}_{(i,j)}$  stores the  $(i,j)$ -th elements of  $\mathbf{R}_{n,\ell}$  associated with all the  $(n, \ell)$  combinations, while  $\mathbf{b} = [r_{0,\ell_1}, \dots, r_{\bar{n},\bar{\ell}}]^\top$ .

**Remark 6.** *Following the GSK scheme, also the PSK iteration is initialized with  $D^0(s_k, \boldsymbol{\vartheta}_m) = 1$ , and the iterations stop when the denominator estimates stabilize.*

The intermediate regressor matrix  $\Psi^\nu$  is defined following (2.25), as

$$\Psi^\nu = \begin{pmatrix} \mathbf{\Gamma}^\nu & & \Xi_{(1,1)}^\nu \\ & \ddots & \vdots \\ & & \mathbf{\Gamma}^\nu & \Xi_{(P,P)}^\nu \end{pmatrix} \in \mathbb{C}^{P^2 K \bar{m} \times (\bar{n}+1)(P^2 |\mathcal{I}_N| + |\mathcal{I}_D|)}, \quad (2.68)$$

with suitable modifications to its building blocks  $\mathbf{\Gamma}^\nu$  and  $\Xi_{(i,j)}^\nu$ , which we define as

$$\mathbf{\Gamma}^\nu = \mathbf{W}^{\nu-1} \mathbf{X}_N, \quad (2.69a)$$

$$\Xi_{(i,j)}^\nu = -\mathbf{W}^{\nu-1} \check{\mathbf{H}}_{(i,j)} \mathbf{X}_D. \quad (2.69b)$$

As in the univariate case, the diagonal matrices  $\mathbf{W}^{\nu-1}$  and  $\check{\mathbf{H}}_{(i,j)}$  collect, respectively, the (inverse) denominator samples  $1/D^{\nu-1}(s_k, \boldsymbol{\vartheta}_m)$  available from the previous iteration, and the sampled frequency responses  $H_{(i,j)}(s_k, \boldsymbol{\vartheta}_m)$ .

In the parameterized setting, the Cauchy matrix  $\Phi$  in (2.69) is replaced by matrix  $\mathbf{X}_{N,D}$ , defined as (the operator  $\otimes$  denotes the *Kronecker* product)

$$\mathbf{X}_{N,D} = \Phi \otimes \mathbf{K}_{N,D} \quad (2.70)$$

whose elements embed the products  $\xi_\ell(\boldsymbol{\vartheta})\varphi_n(s)$  of the frequency/parameter basis functions. The matrix  $\Phi$  is defined as in (2.28) and  $\mathbf{K}_{N,D}$  collect the parameter basis functions, evaluated in correspondence to the available parameter samples, as

$$\mathbf{K}_{N,D} = \begin{pmatrix} \xi_{\ell_1}(\boldsymbol{\vartheta}_1) & \cdots & \xi_{\bar{\ell}}(\boldsymbol{\vartheta}_1) \\ \vdots & & \vdots \\ \xi_{\ell_1}(\boldsymbol{\vartheta}_{\bar{m}_T}) & \cdots & \xi_{\bar{\ell}}(\boldsymbol{\vartheta}_{\bar{m}_T}) \end{pmatrix} \quad (2.71)$$

### Fast-PSK

The PSK iteration will be our algorithm of choice. In view of the further developments of Section 6.1, it is useful to introduce the *Fast-PSK* scheme, originally presented in [54], that dramatically improves the performances of the standard PSK implementation.

Following [55], it is possible to exploit the particular *bordered block diagonal* structure of the matrix  $\Psi^\nu$ , that enables to separate the evaluation of the (matrix) numerator and the (scalar) denominator unknowns. For each input/output port combination  $(i, j)$ , we first evaluate the QR factorizations of the block matrices

$$\mathbf{Q}_{(i,j)}^\nu \mathbf{R}_{(i,j)}^\nu = \begin{pmatrix} \Gamma^\nu & \Xi_{(i,j)}^\nu \end{pmatrix} \quad (2.72)$$

where the upper triangular matrix  $\mathbf{R}_{(i,j)}^\nu$  is partitioned as

$$\mathbf{R}_{(i,j)}^\nu = \begin{pmatrix} \mathbf{R}_{(i,j;1,1)}^\nu & \mathbf{R}_{(i,j;1,2)}^\nu \\ \mathbf{0} & \mathbf{R}_{(i,j;2,2)}^\nu \end{pmatrix} \quad (2.73)$$

Then, when all the sub-matrices  $\mathbf{R}_{(i,j;2,2)}^\nu$ ,  $i, j = 1, \dots, P$ , are available, the intermediate denominator coefficients collected in vector  $\mathbf{b}^\nu$  are computed by

solving

$$\begin{pmatrix} \mathbf{R}_{(1,1;2,2)}^\nu \\ \vdots \\ \mathbf{R}_{(P,P;2,2)}^\nu \end{pmatrix} \mathbf{b}^\nu = \bar{\mathbf{R}}^\nu \mathbf{b}^\nu \approx \mathbf{0} \quad (2.74)$$

As pointed out in [54], this procedure amounts to iteratively solve only for the (reduced) set of denominator unknowns  $\mathbf{b}^\nu$ , which are the only variables required to proceed in the PSK scheme.

Assuming that the iterations stop at  $\nu = \bar{\nu}$ , we finally compute (only once) the numerator unknowns by solving another least squares problem with multiple right-hand sides

$$\mathbf{\Gamma}^{\bar{\nu}} \mathbf{A} \approx \mathbf{B} \quad (2.75)$$

where the matrix unknown is

$$\mathbf{A} = \left( \mathbf{r}_{(1,1)} \quad \dots \quad \mathbf{r}_{(P,P)} \right) \quad (2.76)$$

and

$$\mathbf{B} = \left( -\mathbf{\Xi}_{(1,1)} \mathbf{b}^{\bar{\nu}} \quad \dots \quad -\mathbf{\Xi}_{(P,P)} \mathbf{b}^{\bar{\nu}} \right). \quad (2.77)$$

The advantages of this implementation are particularly evident when the number of ports is large. For a detailed description see the original publication [54].

### Numerical example

We demonstrate the performances of the PSK iteration by constructing a parameterized model of the double-folded microstrip filter depicted in Figure 2.5, first presented in [47, 56]. Its electrical behavior is parameterized through the length  $\vartheta$  of the stubs, that can vary in the range [2.08, 2.28] mm.

This device is characterized by means of its input-output scattering parameters, evaluated in correspondence of a discrete set of  $\bar{m} = 21$  parameter instances in the range [2.08, 2.28] mm, each composed of  $K = 300$  frequency samples, covering the band [5 GHz, 20 GHz]. Setting up the PSK algorithm with  $\bar{n} = 10$ , and  $\mathcal{I}_N = \mathcal{I}_D = \{0, 1, 2\}$ , we obtained a parameterized model of the form (2.63), whose worst-case error with respect to the validation data is

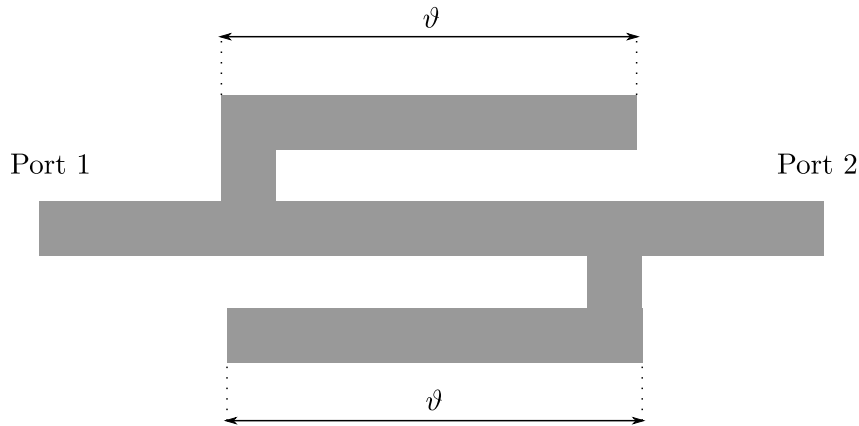


Fig. 2.5 Double-folded microstrip used as test-case for the Parameterized Sanathanan Koerner algorithm.

approximately 1%. Figure 2.6 compares the model responses (red-dashed lines) with respect to the validation data (blue-solid lines).

## Discussion

The standard PSK algorithm is an efficient extension of the GSK scheme to the parameterized setting. Experimental results (see, for instance, [11, 54, 57]) demonstrate the high performances of this method. In addition, the use of a closed model form (2.64) enables to easily convert the frequency-domain PSK model to equivalent parameterized SPICE circuit blocks, to be used as surrogates in time-domain simulations (see [4]). However, a reliable use of these models in circuit solvers may be precluded due to the lack of stability and passivity guarantees.

In [4] and [58], some techniques have been developed to enforce stability and passivity, respectively. Still, these strategies are limited to one parameter only and suffer from some reliability issues, see Section 4.3.4.

Finally, as for the other parameterized modeling approaches, also the PSK can not handle more than few parameters. In fact, for a growing number of independent parameters the memory and CPU requirements blow exponentially (see Section 5.3.2).

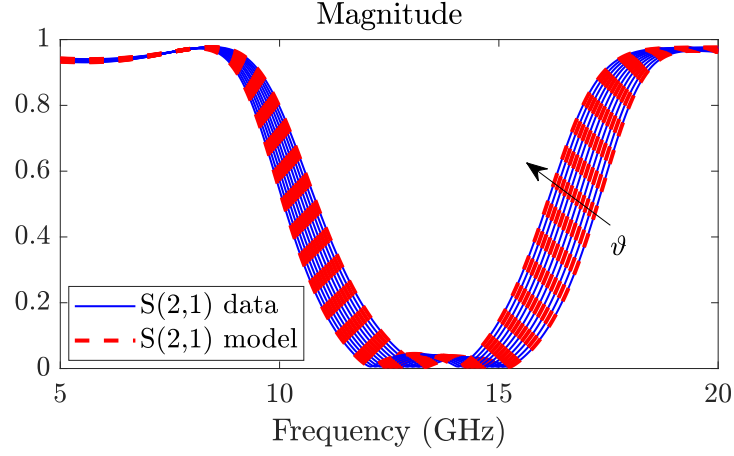


Fig. 2.6 Comparing the parametric raw data (blue-solid line) with the corresponding responses of the parameterized PSK model (red-dashed lines).

### PSK state-space realizations

The PSK model structure can be further expressed in terms of a parameterized state-space realizations, defined as

$$\begin{cases} \dot{\mathbf{x}}(t, \boldsymbol{\vartheta}) = \mathbf{A}(\boldsymbol{\vartheta})\mathbf{x}(t, \boldsymbol{\vartheta}) + \mathbf{B}(\boldsymbol{\vartheta})\mathbf{u}(t) \\ \mathbf{y}(t, \boldsymbol{\vartheta}) = \mathbf{C}(\boldsymbol{\vartheta})\mathbf{x}(t, \boldsymbol{\vartheta}) + \mathbf{D}(\boldsymbol{\vartheta})\mathbf{u}(t) \end{cases} \quad (2.78)$$

Following the procedure in [11], this can be easily achieved by first deriving the state-space realization of the model numerator and denominator separately, as

$$\mathbf{N}(s, \boldsymbol{\vartheta}) \leftrightarrow \left( \begin{array}{c|c} \mathbf{A}_0 & \mathbf{B}_0 \\ \mathbf{C}_1(\boldsymbol{\vartheta}) & \mathbf{D}_1(\boldsymbol{\vartheta}) \end{array} \right) \quad (2.79a)$$

$$D(s, \boldsymbol{\vartheta})\mathbf{I}_p \leftrightarrow \left( \begin{array}{c|c} \mathbf{A}_0 & \mathbf{B}_0 \\ \mathbf{C}_2(\boldsymbol{\vartheta}) & \mathbf{D}_2(\boldsymbol{\vartheta}) \end{array} \right) \quad (2.79b)$$

where, defining the auxiliary matrices ( $\bar{n}_r$  and  $\bar{n}_c$  denote the number of real poles and complex conjugate pairs, respectively)

$$\begin{aligned}
\mathbf{A}_{0,r} &= \text{blkdiag} \{q_n \mathbf{I}_P\}_{n=1}^{\bar{n}_r} & (2.80) \\
\mathbf{A}_{0,c} &= \text{blkdiag} \left\{ \mathbf{I}_P \otimes \begin{pmatrix} q'_n & q''_n \\ -q''_n & q'_n \end{pmatrix} \right\}_{n=\bar{n}_r+1}^{\bar{n}=\bar{n}_r+\bar{n}_c} \\
\mathbf{B}_{0,r} &= \mathbf{I}_P \otimes [1, \dots, 1]^\top \\
\mathbf{B}_{0,c} &= \mathbf{I}_P \otimes [2, 0, \dots, 2, 0]^\top \\
\mathbf{C}_{1,r}(\boldsymbol{\vartheta}) &= [\mathbf{R}_1(\boldsymbol{\vartheta}), \dots, \mathbf{R}_{\bar{n}_r}(\boldsymbol{\vartheta})] \\
\mathbf{C}_{1,c}(\boldsymbol{\vartheta}) &= [\mathbf{R}'_{\bar{n}_r+1}(\boldsymbol{\vartheta}), \mathbf{R}''_{\bar{n}_r+1}(\boldsymbol{\vartheta}), \dots, \mathbf{R}'_{\bar{n}}(\boldsymbol{\vartheta}), \mathbf{R}''_{\bar{n}}(\boldsymbol{\vartheta})] \\
\mathbf{C}_{2,r}(\boldsymbol{\vartheta}) &= [r_1(\boldsymbol{\vartheta})\mathbf{I}_P, \dots, r_{\bar{n}_r}(\boldsymbol{\vartheta})\mathbf{I}_P] \\
\mathbf{C}_{2,c}(\boldsymbol{\vartheta}) &= [r'_{\bar{n}_r+1}(\boldsymbol{\vartheta})\mathbf{I}_P, r_{\bar{n}_r+1}(\boldsymbol{\vartheta})\mathbf{I}_P, \dots, r'_{\bar{n}}(\boldsymbol{\vartheta})\mathbf{I}_P, r''_{\bar{n}}(\boldsymbol{\vartheta})\mathbf{I}_P]
\end{aligned}$$

the realization matrices in (2.79) are defined as

$$\begin{aligned}
\mathbf{A}_0 &= \text{blkdiag}\{\mathbf{A}_{0,r}, \mathbf{A}_{0,c}\} & (2.81) \\
\mathbf{B}_0^\top &= [\mathbf{B}_{0,r}^\top, \mathbf{B}_{0,c}^\top] \\
\mathbf{C}_1(\boldsymbol{\vartheta}) &= [\mathbf{C}_{1,r}(\boldsymbol{\vartheta}), \mathbf{C}_{1,c}(\boldsymbol{\vartheta})] \\
\mathbf{C}_2(\boldsymbol{\vartheta}) &= [\mathbf{C}_{2,r}(\boldsymbol{\vartheta}), \mathbf{C}_{2,c}(\boldsymbol{\vartheta})] \\
\mathbf{D}_1(\boldsymbol{\vartheta}) &= \mathbf{R}_0(\boldsymbol{\vartheta}) \\
\mathbf{D}_2(\boldsymbol{\vartheta}) &= \mathbf{I}_P r_0(\boldsymbol{\vartheta}).
\end{aligned}$$

We remark that  $q_n$  identifies a real pole and  $q'_n, q''_n$  denote the real and imaginary parts of a complex pole. Similarly,  $\mathbf{R}'_n(\boldsymbol{\vartheta}), \mathbf{R}''_n(\boldsymbol{\vartheta})$ , and  $r'_n(\boldsymbol{\vartheta}), r''_n(\boldsymbol{\vartheta})$  identify the real and imaginary parts for the numerator and denominator coefficients, respectively.

A state-space realization for  $\mathbf{H}(s, \boldsymbol{\vartheta}) = \mathbf{N}(s, \boldsymbol{\vartheta})(D(s, \boldsymbol{\vartheta})^{-1}\mathbf{I}_P)$  can be derived as

$$\mathbf{H}(s, \boldsymbol{\vartheta}) \leftrightarrow \left( \begin{array}{c|c} \mathbf{A}_0 - \mathbf{B}_0\mathbf{D}_2^{-1}(\boldsymbol{\vartheta})\mathbf{C}_2(\boldsymbol{\vartheta}) & \mathbf{B}_0\mathbf{D}_2^{-1}(\boldsymbol{\vartheta}) \\ \hline \mathbf{C}_1(\boldsymbol{\vartheta}) - \mathbf{D}_1(\boldsymbol{\vartheta})\mathbf{D}_2^{-1}(\boldsymbol{\vartheta})\mathbf{C}_2(\boldsymbol{\vartheta}) & \mathbf{D}_1(\boldsymbol{\vartheta})\mathbf{D}_2^{-1}(\boldsymbol{\vartheta}) \end{array} \right) \quad (2.82)$$

The above (2.82) is not very practical, since all the realization matrices depend on the parameters in a complex way as a result of the inversion in the



term  $\mathbf{D}_2(\boldsymbol{\vartheta})$ . In [11], it has been shown that this problem can be circumvented resorting to an equivalent descriptor realization

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t, \boldsymbol{\vartheta}) = \mathbf{A}(\boldsymbol{\vartheta})\mathbf{x}(t, \boldsymbol{\vartheta}) + \mathbf{B}\mathbf{u}(t, \boldsymbol{\vartheta}) \\ \mathbf{y}(t, \boldsymbol{\vartheta}) = \mathbf{C}(\boldsymbol{\vartheta})\mathbf{x}(t, \boldsymbol{\vartheta}) \end{cases} \quad (2.83)$$

where

$$\mathbf{A}(\boldsymbol{\vartheta}) = \begin{pmatrix} \mathbf{A}_0 & \mathbf{B}_0 \\ \mathbf{C}_2(\boldsymbol{\vartheta}) & \mathbf{D}_2(\boldsymbol{\vartheta}) \end{pmatrix} \quad \mathbf{E} = \begin{pmatrix} \mathbf{I}_{P\bar{n}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (2.84)$$

$$\mathbf{C}(\boldsymbol{\vartheta}) = (\mathbf{C}_1(\boldsymbol{\vartheta}), \mathbf{D}_1(\boldsymbol{\vartheta})) \quad \mathbf{B} = (\mathbf{0}_{P\bar{n},P}, -\mathbf{I}_P)^\top \quad (2.85)$$

At the cost of a small increase in the number of states, in the descriptor form all the realization matrices are linear in the individual parameterized matrices.

It is remarkable that, in the above descriptor realization, only the matrices  $\mathbf{A}(\boldsymbol{\vartheta})$  and  $\mathbf{C}(\boldsymbol{\vartheta})$  depend on the parameters. In this regard, we highlight that throughout this work it will be often useful to represent the parameterized realization matrices as linear combinations of constant coefficients, weighted by the basis functions  $\xi_\ell(\boldsymbol{\vartheta})$ , as

$$\mathbf{A}(\boldsymbol{\vartheta}) = \sum_{\ell \in \mathcal{I}} \mathbf{A}_\ell \xi_\ell(\boldsymbol{\vartheta}) \quad (2.86)$$

and, similarly, for  $\mathbf{C}(\boldsymbol{\vartheta})$ .

### SPICE synthesis

The PSK model form (2.63) allows for an equivalent representation as a parameterized SPICE netlist [4], to be used as parameterized black-box surrogate in circuit solvers. For the further development we assume the PSK model to be representative of a  $P$ -port circuit block characterized in scattering parameters. Following [4], the SPICE synthesis works as follows. Let us first decompose the rational PSK model into its numerator and denominator components.

1. We consider the denominator as a one-port admittance  $Y_D(s, \boldsymbol{\vartheta}) = D(s, \boldsymbol{\vartheta})$ , with its interface subsidiary voltage  $v_D$  and current  $i_D$ , sat-

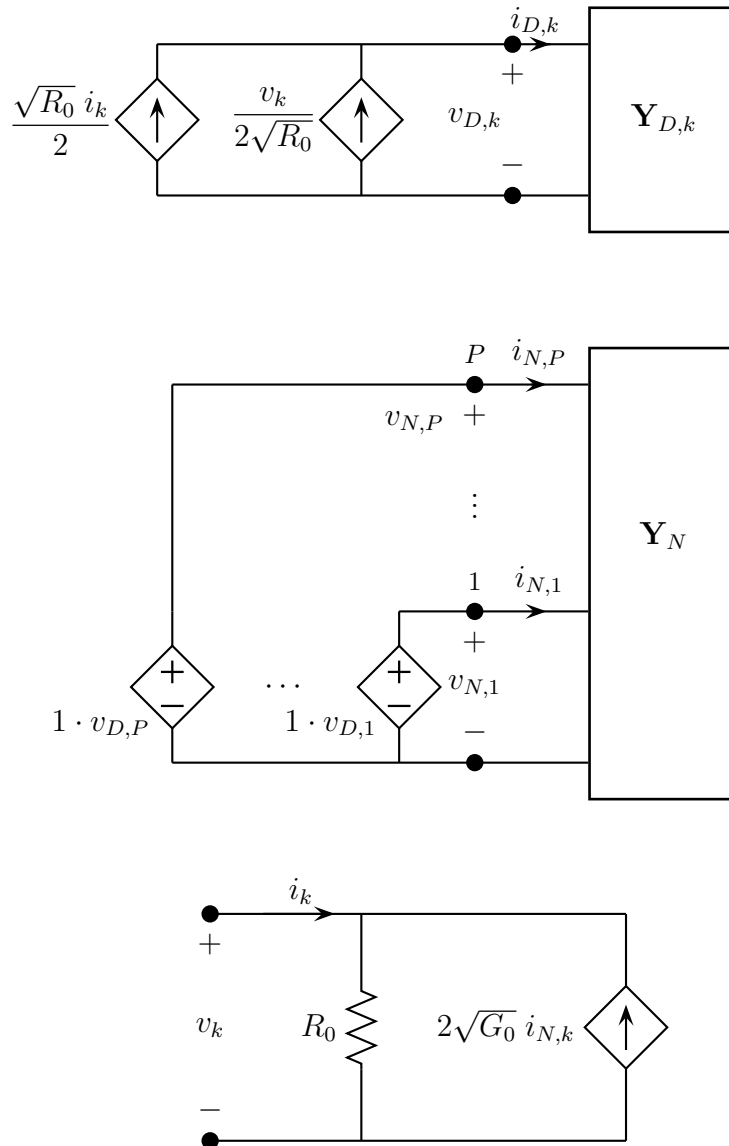


Fig. 2.7 Equivalent SPICE realization of (top panel) denominator sub-block ( $k$ -th out of  $P$ ); (middle panel) numerator sub-block; (bottom panel) external connection of the PSK model ( $k$ -th out of  $P$  in Norton form, scattering representation.)

isfying

$$i_D = Y_D(s, \boldsymbol{\vartheta})v_D \quad (2.87)$$

$$v_D = Y_D^{-1}(s, \boldsymbol{\vartheta})i_D \quad (2.88)$$

The above scalar relations extend to the MIMO case by representing the 1-port denominator as a diagonal  $P$ -port system, as

$$\mathbf{D}(s, \boldsymbol{\vartheta}) = \mathbf{I}_P D(s, \boldsymbol{\vartheta}) \quad (2.89)$$

collecting  $P$  identical replicas of  $D(s, \boldsymbol{\vartheta})$ . Each diagonal element is associated with its interface voltages  $v_{D,k}$  and currents  $i_{D,k}$ , with  $k = 1, \dots, P$ , stored in vectors  $\mathbf{v}_D$  and  $\mathbf{i}_D$ , respectively, obeying

$$\mathbf{i}_D = \mathbf{Y}_D(s, \boldsymbol{\vartheta})\mathbf{v}_D \quad (2.90)$$

$$\mathbf{v}_D = \mathbf{Y}_D^{-1}(s, \boldsymbol{\vartheta})\mathbf{i}_D \quad (2.91)$$

2. The numerator is natively considered as a  $P$ -port admittance  $\mathbf{Y}_N(s, \boldsymbol{\vartheta}) = \mathbf{N}(s, \boldsymbol{\vartheta})$ , that links the subsidiary port currents  $\mathbf{i}_N$  and voltages  $\mathbf{v}_N$ , as

$$\mathbf{i}_N = \mathbf{N}(s, \boldsymbol{\vartheta})\mathbf{v}_N \quad (2.92)$$

3. Under the assumption of scattering representation, the model (2.63) links the incident  $\mathbf{a} = [a_1, \dots, a_P]$  and reflected  $\mathbf{b} = [b_1, \dots, b_P]$  power-normalized waves as

$$\mathbf{b} = \mathbf{H}(s, \boldsymbol{\vartheta})\mathbf{a} = \mathbf{N}(s, \boldsymbol{\vartheta})\mathbf{D}^{-1}(s, \boldsymbol{\vartheta})\mathbf{a} \quad (2.93)$$

where  $a_k, b_k$  are defined as

$$a_k = \frac{1}{2\sqrt{R_0}}(v_k + R_0 i_k) \quad (2.94)$$

$$b_k = \frac{1}{2\sqrt{R_0}}(v_k - R_0 i_k) \quad (2.95)$$

and  $R_0$  identifies the port reference impedance.

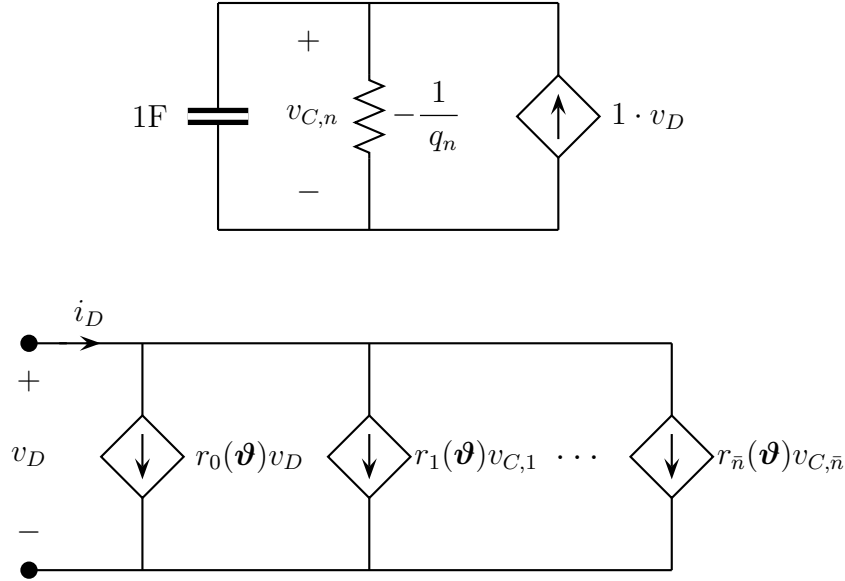


Fig. 2.8 Realization of the denominator admittance sub-block  $D(s, \boldsymbol{\vartheta})$ . Top panel: synthesis of the  $q_n$ -th pole with RC cell; bottom panel: external interface.

These transfer function blocks are interconnected as in Figure 2.7 by employing dependent sources, as

1.  $\mathbf{i}_D = \mathbf{a}$ , so that the voltage measured at the output of the denominator sub-block is  $\mathbf{v}_D = \mathbf{D}(s, \boldsymbol{\vartheta})\mathbf{a}$ . Figure 2.7 (top panel) illustrate how dependent sources are used to synthesize each wave  $a_k$ .
2.  $\mathbf{v}_N = \mathbf{v}_D$ , so that  $\mathbf{b} = \mathbf{H}(s, \boldsymbol{\vartheta})$  (see Figure 2.7, middle panel).
3.  $\mathbf{b} = \mathbf{i}_N$ , to obtain  $\mathbf{b} = \mathbf{H}(s, \boldsymbol{\vartheta})\mathbf{a}$ . See Figure 2.7 (bottom panel), where the output relation (2.95) is synthesized in Norton form.

The synthesis of each transfer function block follows the same approach. In the following, we analyze only the realization of  $D(s, \boldsymbol{\vartheta})$ , the extension to  $\mathbf{D}(s, \boldsymbol{\vartheta})$  and  $\mathbf{N}(s, \boldsymbol{\vartheta})$  is straightforward. From (2.63), it is possible to write

$$i_D = D(s, \boldsymbol{\vartheta})v_D = \sum_{n=0}^{\bar{n}} j_{D,n} \quad (2.96)$$

where  $j_{D,0} = r_0(\boldsymbol{\vartheta})v_D$  and  $j_{D,n} = r_n(\boldsymbol{\vartheta})v_{C,n}$ , with  $v_{C,n} = (s - q_n)^{-1}v_D$ . Under the assumption of stable basis poles, the subsidiary voltages  $v_{C,n}$  are realized as

RC cells (coupled RC cell are used to synthesize complex conjugate basis poles), see Figure 2.8, top panel. Each current  $j_{D,n}$  is realized through a Voltage-Controlled-Current-Source, with a parameterized trans-conductance  $r_n(\boldsymbol{\vartheta})$ . The superposition in (2.96) is obtained as in the bottom panel of Figure 2.8.

### 2.3.4 Other techniques

Beside the above strategies, in the past years other parameterized macromodeling approaches have been proposed. Some rely on parameterized extensions of the univariate Loewner framework, such as [59]. However, as pointed out in [60], this approach works well only if the dynamic order remains constant under parameter variations. Other methods, introduced in [61] and [62] propose to natively build a parameterized version of the Loewner framework. These methods, however, can handle very few parameters due to the curse of dimensionality and, in addition, may give rise to non-smooth behavior of the approximation [63]. These problems have been addressed in [63] where a novel interpretation of the parameterized approximation problem in the Loewner framework has been proposed. We remark that, since all the above approaches exploit a rational barycentric form, it is possible to construct SPICE compatible netlists to be used in circuit solvers.

In addition to the above methods, it has been recently introduced in [64] the p-AAA algorithm, a parameterized extension of the univariate AAA scheme [43]. Despite the good modeling results, also this approach may suffer from the curse of dimensionality when the number of parameters exceeds few units.

The main drawback of all these methods is the lack of stability and passivity enforcement strategies, that undermines the applicability of these models in time-domain simulation settings.

## 2.4 Open problems

To summarize, each parameterized macromodeling algorithm comes with its pros and cons. In regard to the PSK scheme, we already pointed out that

current strategies are not capable of uniformly guaranteeing the models to be stable and passive for more than one independent parameter.

The other major limitation common to *all* the presented macromodeling techniques is the inability of handling more than few independent parameters. This mainly stems from a non-optimized model formulation that, as the number of parameters increases, requires the estimation of an exponentially growing number of model unknowns, making the identification procedure computationally unfeasible for most of the commercial hardware.

First, taking inspiration from [4, 58] we will address the problem of enforcing stability and passivity in Chapter 4. In particular, we are going to propose some relevant improvements to extend their application domain toward high-dimensional setting.

Then, in Chapter 5 we will tackle the problem of high-dimensional modeling. We will propose an optimized model structure built upon unstructured *Radial Basis Functions* capable of handling an increased number of parameters. In addition, we will propose a novel stability enforcement strategy, which is particularly suited in the considered high-dimensional framework.

Finally, in the last Chapter 6 we will propose a fully automated (high-dimensional) modeling flow. The main result of this Chapter is the development of ad-hoc algorithms for the optimization of the hyper-parameters embedded in the model structure.

All the above results will be demonstrated through relevant numerical examples.

# Chapter 3

## Passivity of LTI systems

In this work, we are interested in electric, electronic and electromagnetic structures that are *passive*, in the sense that they are only allowed to either dissipate or redistribute the supplied energy, but not to generate “new” energy on their own. For reasons that will be soon evident, a surrogate model must comply with such energetic properties of the considered device. Unfortunately, the inevitable numerical approximations arising from the modeling procedure (in our case, accomplished through the PSK iteration) may give rise to local violations. When used as a surrogate multi-port subsystem of an interconnected passive network, a non-passive surrogate model may inject nonphysical “*numerical*” energy in the network that, in turn, may drive time-domain simulations to instability. Therefore, the need of reliable and stable simulations is the main driver for requiring the passivity of surrogate macromodels.

In this chapter, we consider the particular case of a LTI parameterized model  $\mathbf{H}(s, \boldsymbol{\vartheta}_0)$ , evaluated in correspondence to a frozen parameter value  $\boldsymbol{\vartheta}_0$ . For simplicity, we refer to this non-parametric model as to  $\mathbf{H}(s)$ . We exploit this particular case to introduce the main results about the theory of dissipativity for LTI systems. The parametric case will be addressed in Chapter 4.

Let us now consider the example of a simple interconnection illustrated in the top-panel of Figure 3.1, composed of a stable two-port surrogate model  $\mathbf{H}(s)$  terminated on a passive load. If the model is not passive for some frequency values, there exists a set of terminations such that, under certain excitation conditions, makes the whole system unstable. The bottom panel of Figure 3.1

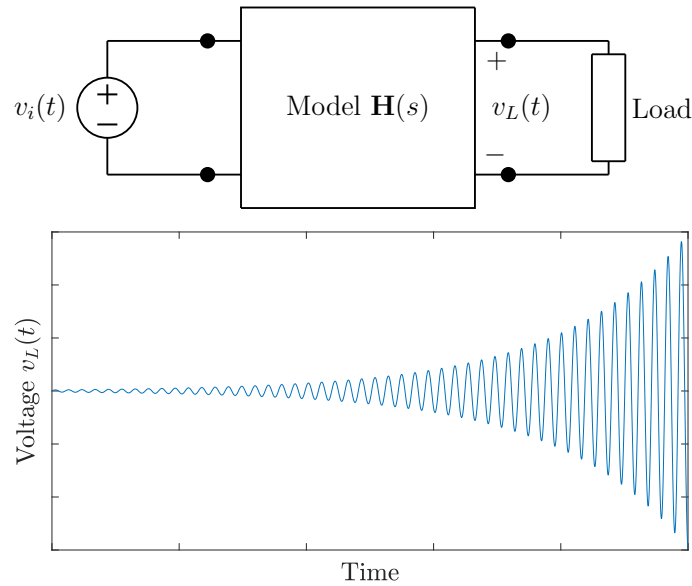


Fig. 3.1 An example of a time-domain simulation for a non-passive macromodel. The top panel depicts the considered system: an interconnection of a non-passive model with a passive load. The system is supplied with a single tone signal  $v_i(t)$ . Bottom panel: resulting unstable voltage  $v_L(t)$ , evaluated at the interconnection node.

depicts the unstable behavior of the load voltage  $v_L(t)$  excited by a single-tone input operating at the frequency where the model behaves as non-passive.

In order to compensate for such non-passive behaviors (from now on referred to as *passivity violations*) it is common to follow a two step approach:

1. first, all the frequency bands where the model  $\mathbf{H}(s)$  exhibits a non-passive behavior are determined, then
2. all the detected passivity violations are removed in a subsequent *enforcement* phase.

The effectiveness of such a *passivity enforcement* scheme strongly depends on the availability of reliable characterization and/or sampling algorithms able to detect *all* the non-passive frequency sub-bands, since unresolved violations can not be removed.

In this Chapter, we first introduce the concept of dissipativity in a general form in Section 3.1. Then, in Sections 3.2 and 3.3 we will focus on LTI systems,



providing a precise passivity characterization both in the time and frequency domain. Finally, in Section 3.4, we address the problem of finding all the passivity violation sub-band introducing Hamiltonian-based adaptive sampling.

## 3.1 Theoretical aspects of dissipativity

In the following, we summarize some known theoretical results associated to the concept of dissipativity. In order to keep the discussion as concise as possible, we will report and analyze only the results of interest, referring the Reader to the cited references for additional details

### 3.1.1 Dissipative systems

Depending on the considered application field, the literature refers to the concept of dissipativity and passivity with different terms. Nevertheless, all the definitions make reference to the *net energy flow* absorbed by the considered multiport system. We introduce the *supply rate* function  $p(t) : U \times Y \rightarrow \mathbb{R}$  that quantifies the amount of *absorbed* power by a generic LTI multi-port system, characterized by  $P$  input  $\mathbf{u}(t) \in U$  and  $P$  output  $\mathbf{y}(t) \in Y$  signals.

Due to practical interests, the literature commonly refers to supply rates associated with *immittance* (either impedance or admittance) and *scattering* representations, for which it holds

1. Immittance:  $p(t) = \mathbf{u}(t)^\top \mathbf{y}(t)$ ,
2. Scattering:  $p(t) = \mathbf{u}(t)^\top \mathbf{u}(t) - \mathbf{y}(t)^\top \mathbf{y}(t)$

More general formulations are available based on the so-called *quadratic supply* [65], defined as

$$p(t) = \mathbf{u}(t)^\top \mathbf{Q} \mathbf{u}(t) + \mathbf{u}(t)^\top \mathbf{S} \mathbf{y}(t) + \mathbf{y}(t)^\top \mathbf{S}^\top \mathbf{u}(t) + \mathbf{y}(t)^\top \mathbf{R} \mathbf{y}(t) \quad (3.1)$$

or, in a more compact form, as

$$p(t) = \begin{pmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) \end{pmatrix}^\top \begin{pmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^\top & \mathbf{R} \end{pmatrix} \begin{pmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) \end{pmatrix} \quad (3.2)$$

where  $\mathbf{Q}$ ,  $\mathbf{S}$ , and  $\mathbf{R}$  are real-valued matrices of dimension  $P \times P$ . The standard *immittance* and *scattering* cases boil down to setting  $\mathbf{Q} = \mathbf{R} = \mathbf{0}$ ,  $\mathbf{S} = \frac{1}{2}\mathbf{I}_P$  and  $\mathbf{Q} = \mathbf{I}_P$ ,  $\mathbf{R} = -\mathbf{I}_P$ ,  $\mathbf{S} = \mathbf{0}$ , respectively.

Based on the concept of supply rate, it is possible to provide [65, 66] the following definition for dissipativity, that correlates the net energy flow with the energy *stored* by the system (for this reason often referred to as *internal passivity*)

**Definition 1.** *A multi-port system is defined dissipative if there exists a scalar storage function  $V(\mathbf{x}(t))$  such that it holds*

$$V(\mathbf{x}(t_2)) \leq V(\mathbf{x}(t_1)) + \int_{t_1}^{t_2} p(t)dt, \quad \forall t_2 \geq t_1 \quad (3.3)$$

where (3.3) is commonly known with the name of *dissipation inequality*, and the scalar function  $V(\mathbf{x}(t)) : X \rightarrow \mathbb{R}$  accounts for the energy stored in the system at any time instant  $t$ . The above definition of dissipativity should be interpreted as follows: the difference in the internal stored energy  $V(\mathbf{x}(t_2)) - V(\mathbf{x}(t_1))$  must not exceed the total energy supplied to the system, identified by the integral term at the right-hand-side. The terms “dissipative” and “passive” are assumed to be synonyms, despite a somewhat inconsistent usage of these two terms in the literature in different application fields of electrical engineering, systems and control, and applied mathematics.

Such a definition of dissipativity implies that a *passive* system can not return power to the external environment before its absorption; this implies, in turn, the system causality. Since we consider only causal physical structures, we adopt this definition for dissipativity. For other (less stringent) definitions of passivity, we refer the Reader to [67, 19].

It is worth noting that, in case the storage function  $V(\mathbf{x})$  is at least differentiable, the dissipation inequality can be stated in differential form, as

$$\frac{dV(\mathbf{x}(t))}{d(t)} \leq p(t) \quad (3.4)$$

In the following, we will use this latter differential interpretation of the dissipation inequality.

## 3.2 Characterization via Linear Matrix Inequalities

The above definitions of dissipativity are fairly general and apply to any linear *and* non-linear system. We now address the particular case of interest in this work, namely the LTI macromodels  $\mathbf{H}(s)$ , that can be realized as a regular state-space system, as

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \quad (3.5)$$

where the model poles are assumed stable and with realization matrices that are real-valued.

For the class of LTI systems, the appropriate choice of storage function is known to be a quadratic function of the states [68],

$$V(\mathbf{x}(t)) = \mathbf{x}(t)^\top \mathbf{P}\mathbf{x}(t) \quad (3.6)$$

with  $\mathbf{P} = \mathbf{P}^\top$ ,  $\mathbf{P} \succ 0$ . Then, plugging in (3.4) the definition of supply rate  $p(t)$  in (3.2) and (3.6), leads to the following passivity condition

$$\frac{d(\mathbf{x}(t)^\top \mathbf{P}\mathbf{x}(t))}{dt} \leq \mathbf{u}(t)^\top \mathbf{Q}\mathbf{u}(t) + \mathbf{u}(t)^\top \mathbf{S}\mathbf{y}(t) + \mathbf{y}(t)^\top \mathbf{S}^\top \mathbf{u}(t) + \mathbf{y}(t)^\top \mathbf{R}\mathbf{y}(t) \quad (3.7)$$

Exploiting the state equations (3.5), the above inequality can be cast in matrix form; we have the following result [65].

**Lemma 1.** *A LTI system, whose supply rate is defined as in (3.2), is passive if and only if there exists  $\mathbf{P} = \mathbf{P}^\top$ ,  $\mathbf{P} \succ 0$ , such that, for any signal  $\mathbf{x}(t)$ ,  $\mathbf{u}(t)$  satisfying the state equations, the following holds*

$$\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}^\top \begin{pmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{C}^\top \mathbf{R}\mathbf{C} & \mathbf{P}\mathbf{B} - (\mathbf{S}\mathbf{C})^\top - \mathbf{C}^\top \mathbf{R}\mathbf{D} \\ \mathbf{B}^\top \mathbf{P} - \mathbf{S}\mathbf{C} - \mathbf{D}^\top \mathbf{R}\mathbf{C} & -\mathbf{Q} - \mathbf{S}\mathbf{D} - (\mathbf{S}\mathbf{D})^\top - \mathbf{D}^\top \mathbf{R}\mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \leq 0 \quad (3.8)$$

The feasibility of the *Linear Matrix Inequality* (LMI) (3.8) provides a dissipativity certificate for the LTI model described by realization matrices

$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ , when the excitation conditions are specified by the supply rate  $p(t)$  through matrices  $\mathbf{Q}, \mathbf{S}, \mathbf{R}$ . In case there are no symmetric *Lyapunov* matrices  $\mathbf{P}$  that satisfy (3.8), the model is not passive.

Although the result of Lemma 1 is general, in literature it is common to split the immittance and scattering characterization. This is easily achieved by selecting appropriately the matrices  $\mathbf{Q}, \mathbf{S}$  and  $\mathbf{R}$  and give rise to the well-known [49, 65] *Positive Real Lemma* (PRL), for immittance parameters, and the *Bounded Real Lemma* (BRL), for scattering parameters

**Lemma 2.** *Positive Real Lemma [49, 65]*

A LTI system in immittance representation, whose states  $\mathbf{x}(t)$  and inputs  $\mathbf{u}(t)$  satisfy the state equations, is passive if and only if there exists  $\mathbf{P} = \mathbf{P}^\top, \mathbf{P} \succ 0$ , such that

$$\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}^\top \begin{pmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} & \mathbf{P} \mathbf{B} - \mathbf{C}^\top \\ \mathbf{B}^\top \mathbf{P} - \mathbf{C} & -(\mathbf{D} + \mathbf{D}^\top) \end{pmatrix} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \leq 0 \quad (3.9)$$

**Lemma 3.** *Bounded Real Lemma [49, 65]*

A LTI system in scattering representation, whose states  $\mathbf{x}(t)$  and inputs  $\mathbf{u}(t)$  satisfy the state equations, is passive if and only if there exists  $\mathbf{P} = \mathbf{P}^\top, \mathbf{P} \succ 0$ , such that

$$\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}^\top \begin{pmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{C}^\top \mathbf{C} & \mathbf{P} \mathbf{B} + \mathbf{C}^\top \mathbf{D} \\ \mathbf{B}^\top \mathbf{P} + \mathbf{D}^\top \mathbf{C} & -(\mathbf{I}_P - \mathbf{D}^\top \mathbf{D}) \end{pmatrix} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \leq 0 \quad (3.10)$$

### 3.3 Passivity characterization in the frequency domain

Beside the algebraic passivity characterization provided by the PRL and the BRL, it is equivalently possible to carry out a passivity characterization in the *Laplace* domain based on the transfer matrix  $\mathbf{H}(s)$  associated to the realization matrices. As a consequence of our hypotheses on the state-space system (3.5), also  $\mathbf{H}(s)$  is assumed to be stable. Introducing first the definitions of *Positive Real* (PR) and *Bounded Real* (BR) transfer matrices [49, 69, 70]

**Definition 2.** *Positive Real Matrix*

A transfer matrix  $\mathbf{H}(s)$  is positive real if the following conditions are concurrently satisfied

1.  $\mathbf{H}(s)$  is analytic in  $\Re\{s\} > 0$ ,
2.  $\mathbf{H}^*(s) = \mathbf{H}(s^*)$ , i.e., the impulse response is real-valued,
3.  $\mathbf{H}(s) + \mathbf{H}^H(s) \succeq 0$ , for  $\Re\{s\} > 0$

**Definition 3.** *Bounded Real Matrix*

A transfer matrix  $\mathbf{H}(s)$  is bounded real if the following conditions are concurrently satisfied

1.  $\mathbf{H}(s)$  is analytic in  $\Re\{s\} > 0$ ,
2.  $\mathbf{H}^*(s) = \mathbf{H}(s^*)$ , i.e., the impulse response is real-valued,
3.  $\mathbf{I} - \mathbf{H}(s)^H \mathbf{H}(s) \succeq 0$ , for  $\Re\{s\} > 0$ .

the connection between the time and the Laplace domain is well-established and stated in the following Theorem [49, 69, 70]

**Theorem 1.** *A LTI system in immittance (resp. scattering) representation associated with a transfer matrix  $\mathbf{H}(s)$  is passive if and only if  $\mathbf{H}(s)$  is Positive Real (resp. Bounded Real).*

Let us analyze more in details the Definitions 2 and 3. The first condition implies that the model  $\mathbf{H}(s)$  is stable, which is satisfied by our assumptions. Also the second condition is structurally guaranteed by the assumption of real-valued realization matrices. On the contrary, there are no a-priori assumptions implying the energy dissipativity entailed by the third condition. So, in order to verify that  $\mathbf{H}(s)$  is either PR or BR (thus, passive), we are required to sample  $\mathbf{I} - \mathbf{H}(s)^H \mathbf{H}(s)$  and  $\mathbf{H}(s) + \mathbf{H}^H(s)$  over the open complex plane  $\mathbb{C}^+$  in order to check their positive semi-definiteness; this is of course impractical.

In our case, we can however introduce some simplifications. It is indeed known [70, 71] that asymptotically stable lumped LTI systems are associated with rational transfer matrices  $\mathbf{H}(s)$ , for which it is possible to check the positive semi-definiteness of  $\mathbf{H}(s) + \mathbf{H}(s)^H$  and  $\mathbf{I} - \mathbf{H}(s)^H \mathbf{H}(s)$  along the only imaginary axis  $s = j\omega$ .

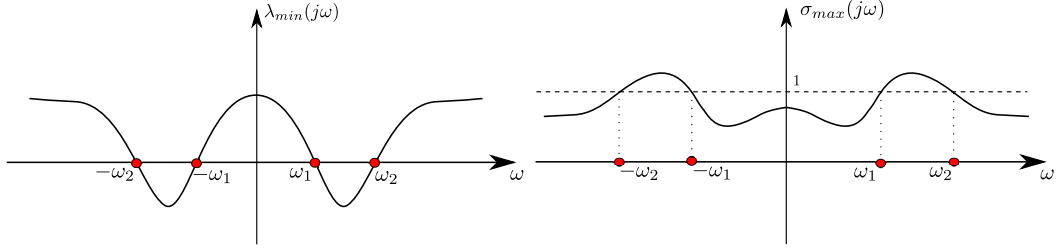


Fig. 3.2 Frequency-domain characterization of passivity. Left panel: immittance model characterized by  $\lambda_{min}(j\omega)$ . Right panel: scattering model characterized by  $\sigma_{max}(j\omega)$ . In both the cases the models are not passive in the frequency sub-band  $[\omega_1, \omega_2]$  (and its respective symmetric  $[-\omega_1, -\omega_2]$ )

Resorting to the following auxiliary quantities

$$\lambda_{min}(j\omega) = \min \lambda \left( \mathbf{H}(j\omega) + \mathbf{H}^H(j\omega) \right) \quad (3.11)$$

$$\sigma_{max}(j\omega) = \max \sigma \left( \mathbf{H}(j\omega) \right)$$

the above positive semi-definiteness conditions can be equivalently stated as

$$\lambda_{min}(j\omega) \geq 0, \quad \forall \omega \quad \text{in case of immittance} \quad (3.12a)$$

$$\sigma_{max}(j\omega) \leq 1, \quad \forall \omega \quad \text{in case of scattering} \quad (3.12b)$$

In the left panel (resp. right panel) of Figure 3.2 we depict the minimum eigenvalue  $\lambda_{min}(j\omega)$  (resp. maximum singular value  $\sigma_{max}(j\omega)$ ) associated with a model in immittance (resp. scattering) representation. Both the models are not passive in the frequency sub-band  $(\omega_1, \omega_2)$ . For completeness, we report also the symmetric part  $\omega < 0$ , which is however redundant due to realness conditions and will not be considered in the following.

The frequency-domain conditions (3.12) provide useful further information regarding the “locations” of passive/non-passive frequency sub-bands, and the worst-case instance of the passivity violations. As we pointed out in the introductory part of this Chapter, in order to guarantee a reliable passivity enforcement procedure, we are required to identify all the non-passive violation sub-bands. Not only, we must also gather the associated worst-case violation (i.e., local minima/maxima of  $\lambda_{min}(j\omega, \boldsymbol{\vartheta})$  or  $\sigma_{max}(j\omega, \boldsymbol{\vartheta})$ ). This procedure requires evaluating samples of  $\lambda_{min}(j\omega)$ ,  $\sigma_{max}(j\omega)$  in correspondence of several discrete frequency values. A possibly adaptive/iterative sampling process in

the frequency domain can be used to retrieve such information. A preferable algebraic approach is discussed next.

### 3.4 Hamiltonian passivity characterization

The identification of the frequency sub-bands corresponding to passivity violations is best achieved through the *Hamiltonian* method. In a nutshell, the Hamiltonian method can be interpreted as an algebraic tool that enables to *exactly* detect the boundaries of passive/non-passive frequency sub-bands ( $\omega_1, \omega_2$  in Figure 3.2), without resorting to expensive sampling techniques. In the following, we will briefly go through the main results; further technical details can be found in [19] and references therein.

Let us collectively define the matrices  $\mathbf{H}(j\omega) + \mathbf{H}(j\omega)^H$  and  $\mathbf{I} - \mathbf{H}(j\omega)^H \mathbf{H}(j\omega)$  as  $\Psi(j\omega)$ , commonly known as *spectral density function* [65]. We flag the frequency value  $j\omega$  as

- *passive (dissipative)*, if  $\Psi(j\omega) \succeq 0$ ,
- *non-passive* otherwise

It is readily understood that transitions between non-passive and passive behaviors occur when (at least) one eigenvalue of  $\Psi(j\omega)$  vanishes. Equivalently, assuming that such a transition occurs at  $j\omega_0$ , there must exist a non-trivial vector  $\mathbf{v} \in \mathcal{N}\{\Psi(j\omega_0)\}$  such that

$$\Psi(j\omega_0)\mathbf{v} = \mathbf{0} \quad (3.13)$$

which implies that the frequency values that bound the passivity violation sub-bands (e.g.,  $\omega_1, \omega_2$  in Figure 3.2) must be sought as the zeros of  $\Psi(j\omega, \boldsymbol{\vartheta})$ .

The evaluation of the spectral density function zeros is better achieved considering its analytic extension  $\Psi(s)$  in the complex plane, and computing the poles of its inverse  $\Psi(s, \boldsymbol{\vartheta})^{-1}$ , to which corresponds the following state-space realization

$$\left( \begin{array}{c|c} \mathbf{A}_\Psi - \mathbf{B}_\Psi \mathbf{D}_\Psi^{-1} \mathbf{C}_\Psi & \mathbf{B}_\Psi \mathbf{D}_\Psi^{-1} \\ \hline -\mathbf{D}_\Psi^{-1} \mathbf{C}_\Psi & \mathbf{D}_\Psi^{-1} \end{array} \right), \quad (3.14)$$

where  $\{\mathbf{A}_\Psi, \mathbf{B}_\Psi, \mathbf{C}_\Psi, \mathbf{D}_\Psi\} = \{\mathbf{A}_\Psi, \mathbf{B}_\Psi, \mathbf{C}_\Psi, \mathbf{D}_\Psi\}$  are the realization matrices of  $\Psi(s)$ . Following standard arguments, the poles of  $\Psi(s, \boldsymbol{\vartheta})^{-1}$  (the zeros of  $\Psi(s, \boldsymbol{\vartheta})$ ) can be thus evaluated as the eigenvalues of  $\mathbf{A}_{\Psi^{-1}} = \mathbf{A}_\Psi - \mathbf{B}_\Psi \mathbf{D}_\Psi^{-1} \mathbf{C}_\Psi$ . Now, the matrix  $\mathbf{A}_{\Psi^{-1}}$  for the two considered input/output representation (immittance or scattering) is easily obtained as

$$\mathbf{M}_I = \begin{pmatrix} \mathbf{A} - \mathbf{B}(\mathbf{D} + \mathbf{D}^\top)^{-1} \mathbf{C} & -\mathbf{B}(\mathbf{D} + \mathbf{D}^\top)^{-1} \mathbf{B}^\top \\ \mathbf{C}^\top (\mathbf{D} + \mathbf{D}^\top)^{-1} \mathbf{C} & -\mathbf{A}^\top + \mathbf{C}^\top (\mathbf{D} + \mathbf{D}^\top)^{-1} \mathbf{B}^\top \end{pmatrix} \quad (3.15)$$

in the immittance case

$$\mathbf{M}_S = \begin{pmatrix} \mathbf{A} + \mathbf{B}(\mathbf{I} - \mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}^\top \mathbf{C} & -\mathbf{B}(\mathbf{I} - \mathbf{D}^\top \mathbf{D})^{-1} \mathbf{B}^\top \\ \mathbf{C}^\top (\mathbf{I} - \mathbf{D}^\top \mathbf{D})^{-1} \mathbf{C} & -\mathbf{A}^\top + \mathbf{C}^\top \mathbf{D} (\mathbf{I} - \mathbf{D}^\top \mathbf{D})^{-1} \mathbf{B}^\top \end{pmatrix} \quad (3.16)$$

in the scattering case. The matrices  $\mathbf{M}_I$  and  $\mathbf{M}_S$  have a *Hamiltonian* structure, for additional details see [19]. Based on this derivation, we have the following result [72]

**Theorem 2.** *Let  $\mathbf{H}(s)$  be the transfer matrix of an immittance (resp. scattering) system associated with realization matrices  $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ , where  $\mathbf{D} + \mathbf{D}^\top$  (resp.  $\mathbf{I} - \mathbf{D}^\top \mathbf{D}$ ) is non-singular and  $\mathbf{A}$  is stable. Then,  $j\omega_0$  is a purely imaginary eigenvalue of  $\mathbf{M}_I$  (resp.  $\mathbf{M}_S$ ) if and only if  $\Psi(j\omega_0)\mathbf{v} = 0$ ,  $\mathbf{v} \neq 0$ .*

For a formal proof, we refer the Reader to [72].

Theorem 2 provides a “practical rule” to infer the passivity properties of LTI systems: instead of resorting to heuristic frequency sampling strategies, the eigenspectrum  $\Lambda$  of suitable Hamiltonian matrices provides a complete dissipativity characterization that is guaranteed not to miss any small localized passivity violation. The frequencies  $\omega_i \in \Omega$  (sorted in ascending order) associated to the purely imaginary Hamiltonian eigenvalues  $j\omega_i \in \Lambda$  partition the frequency axis in disjoint passive and non-passive sub-bands  $\Omega_i = [\omega_{i-1}, \omega_i)$ . The partitioning is completed by adding the left and rightmost bands  $\Omega_0 = [0, \omega_1)$  and  $\Omega_{\bar{i}} = [\omega_{\bar{i}}, \infty)$ .

**Remark 7.** *The Hamiltonian eigenspectrum satisfies a four-quadrant symmetry. Without loss of generality, we can thus consider only the purely imaginary eigenvalues with a positive imaginary part.*



Figure 3.3 provides a graphical illustration of the above partitioning procedure. In the left panel, we are representing the symmetric Hamiltonian eigenspectrum; among the 20 eigenvalues, four  $\{j\omega_1, j\omega_2, -j\omega_1, -j\omega_2\}$  are purely imaginary, of which we consider only the positive subset. Therefore,  $\{\omega_1, \omega_2\}$  induce the frequency-axis partitioning  $[0, \omega_1) \cup [\omega_1, \omega_2) \cup [\omega_2, \infty)$ . Within these sub-bands, Theorem 2 ensures that the system is either dissipative or active; transitions between these two behaviors occur in correspondence of  $\omega = \omega_1$  and  $\omega_2$ , as illustrated in the right panel.

In order to determine whether the considered sub-band  $\Omega_i$  is passive or not, it is sufficient to evaluate  $\lambda_{\min}(j\omega)$  (in the immittance case) or  $\sigma_{\max}(j\omega, \boldsymbol{\vartheta})$  (in the scattering case) for any  $j\omega$  (strictly) within each band  $\Omega_i$ ; as a common practice, we pick the sub-band center  $j\hat{\omega}_i$ . In case  $\lambda_{\min}(j\hat{\omega}_i) > 0$  (or,  $\sigma_{\max}(j\hat{\omega}_i, \boldsymbol{\vartheta}) < 1$ ) the sub-band is uniformly passive (green segment), otherwise it is labelled as active (red segment). For each sub-band  $\Omega_i$  we additionally define the worst-case passivity violation  $\bar{\lambda}_{\min}^i$ , or  $\bar{\sigma}_{\max}^i$ , as

$$\bar{\lambda}_{\min}^i = \arg \min_{\omega \in \Omega_i} \lambda_{\min}(j\omega) \quad (3.17)$$

$$\bar{\sigma}_{\max}^i = \arg \max_{\omega \in \Omega_i} \sigma_{\max}(j\omega) \quad (3.18)$$

attained in correspondence of  $\omega = \bar{\omega}_i$ . The worst-case violations are compactly denoted with the tuples  $(\bar{\omega}_i, \bar{\lambda}_{\min}^i)$  or  $(\bar{\omega}_i, \bar{\sigma}_{\max}^i)$ .

The above Hamiltonian matrices can be thus employed as a powerful tool to assess the passivity of LTI models, as stated by the following [18, 19]

**Theorem 3.** *Let  $\mathbf{H}(s)$  be the transfer matrix of an immittance (resp. scattering) system described by state-space matrices  $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ , where  $\mathbf{D} + \mathbf{D}^\top \succ 0$  (resp.  $\mathbf{I} - \mathbf{D}^\top \mathbf{D} \succ 0$ ) and  $\mathbf{A}$  is stable. If the Hamiltonian matrix  $\mathbf{M}_I$  (resp.  $\mathbf{M}_S$ ) does not have purely imaginary eigenvalues, then  $\mathbf{H}(s)$  is passive.*

### Extended Hamiltonian pencil

In Theorem 2 we deliberately left out the cases where  $(\mathbf{D} + \mathbf{D}^\top)$  and  $(\mathbf{I} - \mathbf{D}^\top \mathbf{D})$  are singular, for which the Hamiltonian matrices  $\mathbf{M}_I$ ,  $\mathbf{M}_S$  are not defined. Under these conditions, a more general approach based on the so-called *Skew-Hamiltonian Hamiltonian* (SHH) pencils is available. In particular, slightly

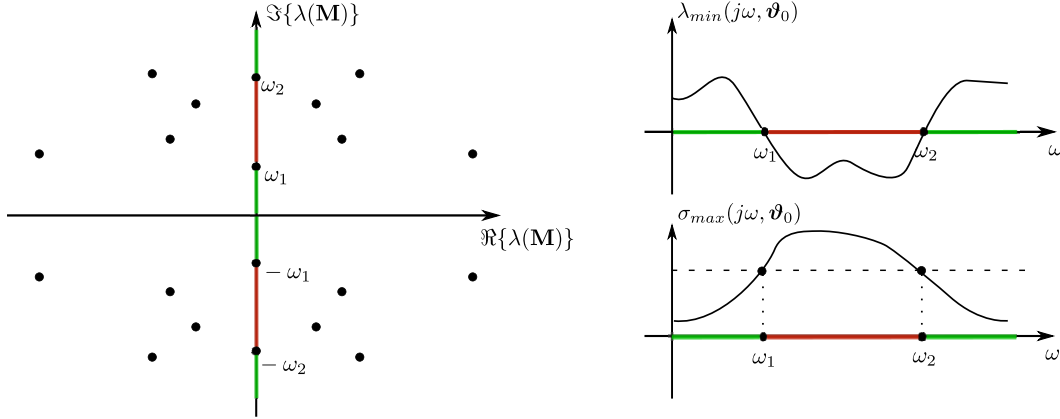


Fig. 3.3 Illustration of the Hamiltonian-based passivity characterization. Right panel: symmetric Hamiltonian eigenspectrum. Left panel: representation of  $\lambda_{\min}(j\omega)$  and  $\sigma_{\max}(j\omega)$  as a function of the frequency. The purely imaginary eigenvalues  $\omega_1, \omega_2$  partition the frequency axis in passive (green segments) and non-passive (red segments) sub-bands.

modifying the proof that leads to Theorem 2 [73] it is possible to show that the following equalities hold

$$\underbrace{\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^\top & -\mathbf{C}^\top \\ \mathbf{C} & \mathbf{B}^\top & \mathbf{D} + \mathbf{D}^\top \end{pmatrix}}_{\mathbf{M}_{I,ext}} \mathbf{q} = j\omega_0 \underbrace{\begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{\mathbf{Y}_I} \mathbf{q} \quad (3.19)$$

for the immittance, and

$$\underbrace{\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}^\top & \mathbf{0} & -\mathbf{C}^\top \\ \mathbf{0} & \mathbf{B}^\top & -\mathbf{I} & \mathbf{D}^\top \\ \mathbf{C} & \mathbf{0} & \mathbf{D} & \mathbf{I} \end{pmatrix}}_{\mathbf{M}_{S,ext}} \mathbf{q} = j\omega_0 \underbrace{\begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{\mathbf{Y}_S} \mathbf{q} \quad (3.20)$$

for the scattering representation, respectively, whenever one eigenvalue of  $\Psi(j\omega)$  vanishes in correspondence of  $j\omega = j\omega_0$ . The above can be regarded as *Generalized Eigenvalue Problems* (GEP) associated with the SHH pairs  $(\mathbf{M}_{I,ext}, \mathbf{Y}_I)$  and  $(\mathbf{M}_{S,ext}, \mathbf{Y}_S)$  whose purely imaginary solutions  $j\omega_0$  correspond to the spectral zeros of  $\Psi(s)$ .

Thus, in case the terms  $(\mathbf{D} + \mathbf{D})$  and  $(\mathbf{I} + \mathbf{D}^\top \mathbf{D})$  are singular, the set of non-passive sub-bands (if any) can be computed resorting to the computation of the generalized SHH (finite) eigenvalues, associated with GEPs (3.19) and (3.20).

**Remark 8.** *The SHH pencils  $(\mathbf{M}_{I,ext}, \mathbf{Y}_I)$  and  $(\mathbf{M}_{S,ext}, \mathbf{Y}_S)$  are characterized by the presence of  $P$  and  $2P$  infinite eigenvalues, respectively, corresponding to the dimension of the singular subspace of matrices  $\mathbf{Y}_I$  and  $\mathbf{Y}_S$ . This justifies why we refer to the subset of finite SHH eigenvalues.*

### Generalized Hamiltonian pencils

As a further generalization, we consider the parameterized model  $\mathbf{H}(s)$  in the following descriptor form

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \quad (3.21)$$

which is assumed to be regular and impulse free. As in the case of regular state-space models, following similar arguments it is possible to prove that the zeros of the spectral density function  $\Psi(s)$  associated to a model in descriptor form correspond to the generalized finite eigenvalues of a SHH matrix pair. In details,  $\Psi(j\omega_0)$  is singular if and only if  $j\omega_0$  satisfies the following generalized eigenvalue problems

$$\underbrace{\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^\top & -\mathbf{C}^\top \\ \mathbf{C} & \mathbf{B}^\top & \mathbf{0} \end{pmatrix}}_{\mathbf{M}_{I,ext}} \mathbf{q} = j\omega_0 \underbrace{\begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{\mathbf{Y}_I} \mathbf{q} \quad (3.22)$$

for the immittance, and

$$\underbrace{\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}^\top & \mathbf{0} & -\mathbf{C}^\top \\ \mathbf{0} & \mathbf{B}^\top & -\mathbf{I} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}}_{\mathbf{M}_{S,ext}} \mathbf{q} = j\omega_0 \underbrace{\begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{\mathbf{Y}_S} \mathbf{q} \quad (3.23)$$

for the scattering representation.

For ease of explanation, in the following we will always refer to the more general case of Skew-Hamiltonian Hamiltonian GEP problem, with the associated set of (finite) generalized eigenvalues. This choice is not restrictive, since the standard eigenvalue problem associated with matrices  $\mathbf{M}_I$ ,  $\mathbf{M}_S$  can be trivially converted to an equivalent GEP as

$$\mathbf{M}_I \rightarrow (\mathbf{M}_I, \mathbf{I}) \quad (3.24)$$

$$\mathbf{M}_S \rightarrow (\mathbf{M}_S, \mathbf{I}) \quad (3.25)$$

### 3.5 Enforcing passivity

We now address the problem of *enforcing* passivity of non-passive models. The passivity enforcement problem can be summarized as follows: given an initially non-passive model  $\mathbf{H}(s)$ , find a suitably small perturbation  $\Delta\mathbf{H}(s)$  such that the perturbed model

$$\widehat{\mathbf{H}}(s) = \mathbf{H}(s) + \Delta\mathbf{H}(s) \quad (3.26)$$

is uniformly passive over  $j\omega$  and, concurrently accurate over the band of interest. Hence, enforcing the model passivity amounts to solving the following optimization problem

$$\begin{aligned} \min \|\Delta\mathbf{H}(s)\| \\ \text{s.t. } \widehat{\mathbf{H}}(s) \text{ is passive} \end{aligned} \quad (3.27)$$

Expressing the perturbation  $\Delta\mathbf{H}(s)$  in terms of its descriptor realization matrices, as

$$\Delta\mathbf{H}(s) = \Delta\mathbf{C} (s\mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \quad (3.28)$$

where we follow the common choice to perturb only the state-output matrix  $\Delta\mathbf{C}$ . In fact, it is not advisable to perturb the matrices  $\mathbf{E}$ ,  $\mathbf{A}$ , in order to preserve poles. Under these assumptions, the model perturbation  $\Delta\mathbf{H}(s)$  is linear in the decision variable  $\Delta\mathbf{C}$  leading, in turn, to a convex cost function.

The dissipativity conditions we discussed in the previous sections will be regarded as constraints that must be enforced to guarantee the model passivity. We can formulate three alternative passivity constraints.

### 3.5.1 LMI constraints

Let us assume that  $\mathbf{H}(s)$  is an asymptotically stable *non*-passive model, associated to state-space realization matrices  $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ .<sup>1</sup> In addition, we assume the model to be *asymptotically passive*, i.e., it is passive at  $s = j\omega \rightarrow \infty$ . The PRL and BRL (see Section 3.2) guarantee that there is not a symmetric and positive definite matrix  $\mathbf{P}$  satisfying

$$\underbrace{\begin{pmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} & \mathbf{P}^\top \mathbf{B} - \mathbf{C}^\top \\ \mathbf{B}^\top \mathbf{P} - \mathbf{C} & -(\mathbf{D} + \mathbf{D}^\top) \end{pmatrix}}_{\mathbf{L}_I(\mathbf{P})} \preceq 0 \quad \underbrace{\begin{pmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} & \mathbf{P} \mathbf{B} & \mathbf{C}^\top \\ \mathbf{B}^\top \mathbf{P} & -\mathbf{I} & \mathbf{D}^\top \\ \mathbf{C} & \mathbf{D} & -\mathbf{I} \end{pmatrix}}_{\mathbf{L}_S(\mathbf{P})} \preceq 0 \quad (3.29)$$

for immittance ( $\mathbf{L}_I(\mathbf{P})$ ) and scattering ( $\mathbf{L}_S(\mathbf{P})$ ) models. The equivalence of (3.10) and  $\mathbf{L}_S(\mathbf{P})$  (which is preferable since the realization matrices appear linearly) is verified by computing the Schur complement of the bottom-right block of  $\mathbf{L}_S(\mathbf{P})$ . In the following, both the above matrices will be identified as  $\mathbf{L}(\mathbf{P})$ .

Provided that the perturbation  $\Delta \mathbf{C}$  on the state-output matrix accordingly perturb the matrix  $\mathbf{L}$  as  $\mathbf{L} + \Delta \mathbf{L}$ , following [75, 76] passivity can be enforced solving

$$\min \|\Delta \mathbf{H}\| \quad \text{s.t. } \mathbf{P} = \mathbf{P}^\top \succ 0, \mathbf{L}(\mathbf{P}) + \Delta \mathbf{L}(\mathbf{P}) \preceq 0 \quad (3.30)$$

with  $\Delta \mathbf{H} = \Delta \mathbf{C} (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B}$ . The above optimization problem is a convex LMI-constrained problem, routinely solved with off-the-shelf solvers. Although the theoretical guarantee that a unique solution exists, the huge CPU effort required to solve (3.30) makes this approach feasible only for reduced size problems [77].

<sup>1</sup>We consider a state-space realization, since the formulation of LMI constraints based on descriptor forms necessitate to address some additional technical questions, which are deemed outside the scope of this work (see [74]).

### 3.5.2 Hamiltonian perturbation

Assuming that  $\mathbf{H}(s)$  is an asymptotically stable model, the purely imaginary eigenvalues  $j\omega_i$  of the associated Hamiltonian matrix  $\mathbf{M}$  (see Section 3.4) denote the presence of non-passive frequency bands. The Hamiltonian perturbation scheme [18] envisages to enforce passivity by inducing spectral perturbations on the purely imaginary eigenvalues of  $\mathbf{M}$ , so that they are displaced outside the imaginary axis in order to remove passivity violations.

Considering a perturbation of state-output matrix as  $\mathbf{C} + \Delta\mathbf{C}$ , the Hamiltonian matrix is correspondingly perturbed as  $\mathbf{M} + \Delta\mathbf{M}$ ; in turn, the purely imaginary eigenvalues  $j\omega_i$  are modified as  $j\omega_i + j\Delta\omega_i(\Delta\mathbf{C})$ . The passivity enforcement procedure devised in [18] amounts to finding the minimal perturbations  $\Delta\mathbf{C}$  so that the extent of  $\Delta\omega_i(\Delta\mathbf{C})$  is sufficient to move the eigenvalues  $j\omega_i$  out of the imaginary axis. In [18] it has been shown that, under a first-order approximation of  $\Delta\omega_i(\Delta\mathbf{C})$ , this problem can be cast as a linearly constrained least-squares, whose solution is standard. The linearization procedure makes the considered perturbation approach valid only up to first-order approximation, hence multiple iterations may be required to compensate all the passivity violations. Some advanced implementations of this approach can be found in [77, 78].

### 3.5.3 Local frequency constraints

Consider once again a non-passive and asymptotically stable model, for which all the violation sub-bands  $\Omega_i = [\omega_{i-1}, \omega_i)$ , together with the worst-case violations  $(\bar{\omega}_i, \bar{\lambda}_{\min}^i)$ , or  $(\bar{\omega}_i, \bar{\sigma}_{\min}^i)$ , are known (for instance, by means of the Hamiltonian sampling outlined in Section 3.4). In Figure 3.4 we depict the least eigenvalue  $\lambda_{\min}(j\omega)$  (black solid line), together with the associated worst-case violations  $\bar{\lambda}_{\min}^1, \bar{\lambda}_{\min}^2$  for an illustrative immittance model (similar conclusions hold for scattering models, considering the largest singular value  $\sigma_{\max}(j\omega)$ ).

Assuming to perturb the state-output matrix as  $\mathbf{C} + \Delta\mathbf{C}$ , the least eigenvalue  $\lambda_{\min}(j\omega)$  or the largest singular value  $\sigma_{\max}(j\omega)$  are correspondingly modified. As introduced in [79, 80], to enforce passivity we seek for a minimal perturbation

$\Delta \mathbf{C}$  that locally displaces all the local minima  $\bar{\lambda}_{\min}^i$  or maxima  $\bar{\sigma}_{\max}^i$  such that

$$\bar{\lambda}_{\min}^i + \Delta \bar{\lambda}_{\min}^i(\Delta \mathbf{C}) \geq 0, \quad \text{or} \quad \bar{\sigma}_{\max}^i + \Delta \bar{\sigma}_{\max}^i(\Delta \mathbf{C}) \leq 1. \quad (3.31)$$

Expressing the perturbations  $\Delta \bar{\lambda}_{\min}^i(\Delta \mathbf{C})$  and  $\Delta \bar{\sigma}_{\max}^i(\Delta \mathbf{C})$  through a first-order linearization, the above (3.31) are used to formulate linear passivity constraints of the following optimization problem

$$\min \|\Delta \mathbf{C}\| \quad \text{s.t.} \quad \mathbf{g}_i^T \cdot \mathbf{x} - h_i \leq 0, \quad i = 1, \dots, \bar{i} \quad (3.32)$$

where  $\mathbf{g}_i$  collect suitably computed coefficient and  $\mathbf{x}$  stores the vectorized decision variable, respectively (see [79, 80] for details). The scalar  $h_i$  is set to  $\bar{\lambda}_{\min}^i$  and  $1 - \bar{\sigma}_{\max}^i$  for the immittance and scattering cases, respectively. The dashed line in Figure 3.4 shows the passive  $\lambda_{\min}(j\omega)$  as resulting from (3.32).

This will be our method of choice for passivity enforcement of parameterized models, therefore a detailed derivation for the above expressions will be provided later in the dedicated Section 4.4.

Solving (3.32) does not necessarily lead to a passive model. In fact, the linear approximation of the real eigenvalues/singular values perturbations induced by  $\Delta \mathbf{C}$  provide only a simplified polytopic representation of the “real” feasibility set of (3.27). Nonetheless, the linearized constraints provide a reduction of the violation regions; therefore, the problem (3.32) is repeatedly solved until all the violations are eliminated.

**Remark 9.** *We remark that the enforcement of local passivity constraints may induce the onset of unwanted violations in already passive sub-bands. This problem has been tackled in [81] where a robust scheme, collecting the constraints associated to subsequent iterations, prevents the appearance of unexpected violations.*

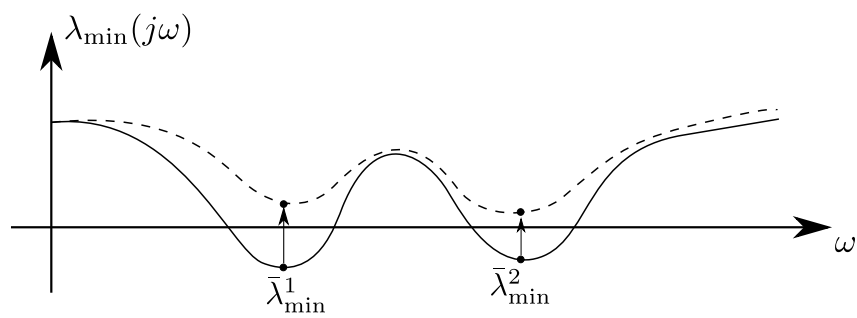


Fig. 3.4 Passivity enforcement via local frequency constraints. Black solid line: least eigenvalue of the original non-passive model. Black dashed line: least eigenvalue of the perturbed passive model.



# Chapter 4

## Checking and enforcing passivity of parameterized models

In this Chapter, we are going to elaborate on the dissipativity characterization discussed in Chapter 3 to extend its applicability to the parameterized framework. For the further developments, in Section 4.1 we first introduce a scheme to enforce stability, uniformly over the parameter space. Then, in Section 4.2 we introduce the problem of checking passivity of parameterized models, for which we propose an innovative solution in Section 4.3, further used to set up a passivity enforcement scheme in Section 4.4. The results are reported in Section 4.5. Finally, limitations of the method and its application to uniform stability enforcement are discussed in Sections 4.6 and 4.7.

### 4.1 Parametric uniform model stability

We recall from Section 2.3.3, that the PSK parameterized model that we consider has the form (2.64), that we report below for convenience

$$\mathbf{H}(s, \boldsymbol{\vartheta}) = \frac{\sum_{\ell \in \mathcal{I}_N} \sum_{n=0}^{\bar{n}} \mathbf{R}_{n,\ell} \xi_{\ell}(\boldsymbol{\vartheta}) \varphi_n(s)}{\sum_{\ell \in \mathcal{I}_D} \sum_{n=0}^{\bar{n}} r_{n,\ell} \xi_{\ell}(\boldsymbol{\vartheta}) \varphi_n(s)} = \mathbf{P}_0(\boldsymbol{\vartheta}) + \sum_{n=1}^{\bar{n}} \frac{\mathbf{P}_n(\boldsymbol{\vartheta})}{s - p_n(\boldsymbol{\vartheta})} \quad (4.1)$$

to which are associated the parameterized state-space and descriptor realizations

$$\underbrace{\left( \begin{array}{c|c} \mathbf{A}(\boldsymbol{\vartheta}) & \mathbf{B}(\boldsymbol{\vartheta}) \\ \hline \mathbf{C}(\boldsymbol{\vartheta}) & \mathbf{D}(\boldsymbol{\vartheta}) \end{array} \right)}_{\text{state-space}} \quad \underbrace{\left( \begin{array}{c|c} [\mathbf{A}(\boldsymbol{\vartheta}), \mathbf{E}] & \mathbf{B} \\ \hline \mathbf{C}(\boldsymbol{\vartheta}) & \mathbf{0} \end{array} \right)}_{\text{descriptor}} \quad (4.2)$$

defined in (2.78) and (2.83), respectively. We have the following definition:

**Definition 4** (Uniform stability). *A parameterized LTI model of the form (4.1) is defined uniformly (asymptotically) stable if  $\Re\{p_n(\boldsymbol{\vartheta})\} < 0 \forall n$  and  $\forall \boldsymbol{\vartheta} \in \Theta$ . Equivalently, a parameterized LTI model in state-space (resp. descriptor) realization is uniformly (asymptotically) stable if the eigenvalues of  $\mathbf{A}(\boldsymbol{\vartheta})$  (resp. of the pencil  $(\mathbf{A}(\boldsymbol{\vartheta}), \mathbf{E})$ ) have a uniformly negative real part  $\forall \boldsymbol{\vartheta} \in \Theta$ .*

As already mentioned in Chapter 2, the parameterized model poles never appear explicitly in the Parameterized Sanathanan-Koerner scheme that, instead, provides an implicit parameterization. Hence, we lose the direct control over the model poles that, for some parameter instances, may become unstable and a direct stability enforcement during the fitting step is thus unfeasible.

The strategy proposed in [58] considers a different viewpoint: instead of considering the model poles, the uniform model stability is attained by enforcing the model denominator to be a *Positive Real* (PR) function [49, 70]. Following Definition 2 for Positive Real Matrices, the *scalar* denominator function

$$D(s, \boldsymbol{\vartheta}) = \sum_{\ell \in \mathcal{I}_D} r_{0,\ell} \xi_\ell(\boldsymbol{\vartheta}) + \sum_{n=1}^{\bar{n}} \sum_{\ell \in \mathcal{I}_D} r_{n,\ell} \frac{\xi_\ell(\boldsymbol{\vartheta})}{s - q_n} \quad (4.3)$$

is defined PR if, for all  $\boldsymbol{\vartheta} \in \Theta$ , it holds that

1.  $D(s, \boldsymbol{\vartheta})$  is regular for  $\Re\{s\} > 0$ ;
2.  $D^*(s, \boldsymbol{\vartheta}) = D(s^*, \boldsymbol{\vartheta})$ ;
3.  $\Re\{D(s, \boldsymbol{\vartheta})\} > 0$  for  $\Re\{s\} > 0$ .

Note that, if the denominator is PR, also  $D^{-1}(s, \boldsymbol{\vartheta})$  is PR. Provided that any PR system is stable [70] since regular for  $\Re\{s\} > 0$ , we conclude that the denominator positive realness (PR-ness) is a sufficient condition for uniform stability.

Based on the considerations of Section 3.3, the validity of the first two conditions is always verified by our assumptions on the model structure, whereas we have no guarantees about the validity of the third, that depends on the actual values of the denominator coefficients. Thus, in order to guarantee the PR-ness of the denominator (thus, uniform stability over  $\Theta$ ), we must enforce the third condition uniformly over the continuous domain  $\mathbb{C}^+ \times \Theta$

#### 4.1.1 Enforcing uniform stability via linear constraints

Based on the above considerations, it is possible to formulate a set of (strict) positive realness constraints for  $D(s, \boldsymbol{\vartheta})$  can be formulated as

$$\Re \left\{ \sum_{n=0}^{\bar{n}} \sum_{\ell \in \mathcal{I}_D} r_{n,\ell} \xi_{\ell}(\hat{\boldsymbol{\vartheta}}_q) \varphi_n(j\hat{\omega}_q) \right\} > \gamma > 0 \quad (4.4)$$

over a discrete set of frequency-parameter combinations  $(j\hat{\omega}_q, \hat{\boldsymbol{\vartheta}}_q)$ , where  $\gamma$  is an arbitrary positive constant. The above PR-ness constraints can be easily enforced during the model fitting procedure. Therefore, at each PSK iteration  $\nu$ , we first collect the locations  $(j\hat{\omega}_q, \hat{\boldsymbol{\vartheta}}_q)$  in set  $\mathcal{A}^\nu$ , then we compute the denominator unknowns solving the following *stability preserving* optimization problem (here reported in the *Fast-PSK form*)

$$\min \left\| \bar{\mathbf{R}}^\nu \mathbf{b}^\nu \right\| \quad (4.5)$$

$$\text{s.t. } \mathbf{P}^\nu \mathbf{b}^\nu > \mathbf{d} \quad (4.6)$$

where  $\mathbf{P}_{(q;n,\ell)} = r_{n,\ell}^\nu \xi_{\ell}(\hat{\boldsymbol{\vartheta}}_q) \varphi_n(j\hat{\omega}_q)$  and the  $q$ -th element of  $\mathbf{d}$  is  $\gamma$ . The numerator coefficients are then evaluated following the PSK scheme.

Concerning the selection of the frequency-parameter combinations  $(j\hat{\omega}_q, \hat{\boldsymbol{\vartheta}}_q)$ , the most straightforward choice is to pick the samples  $(j\hat{\omega}_q, \hat{\boldsymbol{\vartheta}}_q)$  over a dense grid over  $j\mathbb{R} \times \Theta$ . This approach has two main drawbacks:

- it does not provide any guarantee not to miss localized frequency-parameter combinations where the denominator is not PR, and
- the resulting number of constraints quickly blows up when the number of parameters exceed few units.

An optimized strategy for selecting a reduced, yet sufficient, set of PR-ness constraints will be provided in Section 4.7. This strategy requires however a number of preliminary theoretical and algorithmic tools, discussed next. The above uniform stability procedure has been introduced as an anticipation, since the uniform stability will be a prerequisite for all subsequent derivations.

## 4.2 Parametric uniform passivity

Let us now turn to extend the passivity characterization of Chapter 3 to the parameterized setting. The availability of the above discussed uniform stability enforcement scheme enables for a straightforward extension of the uni-variate dissipativity description. In particular, based on (3.11), we introduce the auxiliary quantities

$$\begin{aligned}\lambda_{\min}(\mathbf{j}\omega, \boldsymbol{\vartheta}) &= \min \lambda \left( \mathbf{H}(\mathbf{j}\omega, \boldsymbol{\vartheta}) + \mathbf{H}^{\mathbf{H}}(\mathbf{j}\omega, \boldsymbol{\vartheta}) \right) \\ \sigma_{\max}(\mathbf{j}\omega, \boldsymbol{\vartheta}) &= \max \sigma \left( \mathbf{H}(\mathbf{j}\omega, \boldsymbol{\vartheta}) \right)\end{aligned}\quad (4.7)$$

and we formulate the following definition

**Definition 5** (Uniform passivity). *A uniformly stable parameterized LTI model of the form (4.1) is defined uniformly passive if*

$$\lambda_{\min}(\mathbf{j}\omega, \boldsymbol{\vartheta}) \geq 0, \quad \forall \omega, \boldsymbol{\vartheta} \in \Theta \quad \text{in case of immittance} \quad (4.8a)$$

$$\sigma_{\max}(\mathbf{j}\omega, \boldsymbol{\vartheta}) \leq 1, \quad \forall \omega, \boldsymbol{\vartheta} \in \Theta \quad \text{in case of scattering} \quad (4.8b)$$

In contrast with the uni-variate framework, where the Hamiltonian scheme allows to algebraically pinpoint all the non-passive frequency sub-bands, in the parameterized setting checking uniform passivity is much more complicated. In fact, we are now required to sample a  $(\rho+1)$ -dimensional space in order to detect all the areas (Fig. 4.1 left panel, red regions), volumes (Fig. 4.1, right panel, red regions) and, in general, higher dimensional manifolds that incorporate the frequency-parameter combinations  $(\mathbf{j}\omega, \boldsymbol{\vartheta})$  for which the conditions (4.8) are violated.

Based on the results of Chapter 3, we can however introduce some simplifications. Indeed, if we employ the Hamiltonian scheme, instead of sampling the

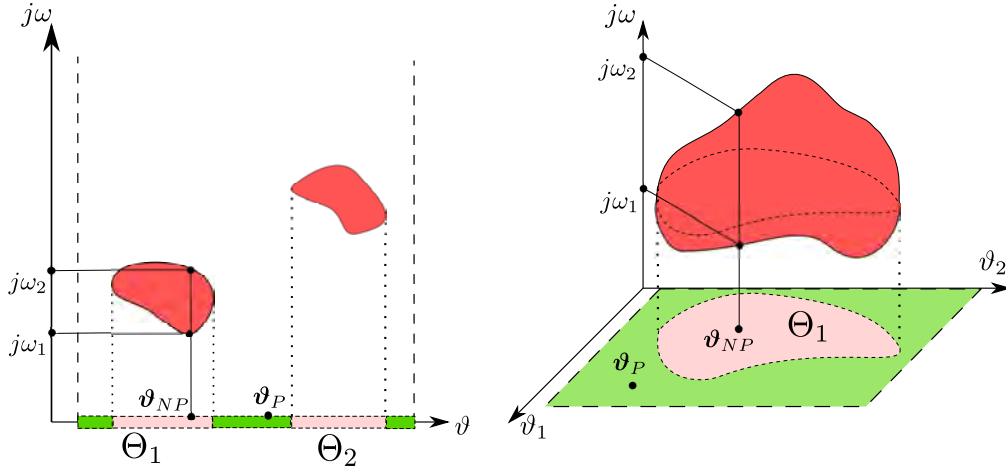


Fig. 4.1 The left and right panels show a graphical illustration of the violation areas  $\Theta_\alpha$  for a one- and bi-dimensional parameter space, respectively.

$(\rho + 1)$ -dimensional frequency-parameter space, we can restrict our search to the smaller parameter space only; here, it is required to detect (and accurately track) all the disjoint regions  $\Theta_\alpha \in \Theta$ ,  $\alpha = 1, 2, \dots$  (represented as light red segments and surfaces in the left and right panels of Figure 4.1, respectively) where the dissipativity conditions (4.8) are not satisfied.

For each of these regions, we define the worst-case passivity violation based on the definition (3.11), as

$$\bar{\lambda}_{\min}^\alpha = \inf_{\omega \in \mathbb{R}, \boldsymbol{\vartheta} \in \Theta_\alpha} \lambda_{\min}(j\omega, \boldsymbol{\vartheta}) \quad (4.9)$$

$$\bar{\sigma}_{\max}^\alpha = \sup_{\omega \in \mathbb{R}, \boldsymbol{\vartheta} \in \Theta_\alpha} \sigma_{\max}(j\omega, \boldsymbol{\vartheta}) \quad (4.10)$$

for immittance and scattering systems, respectively. Together with their location  $(\bar{\omega}_\alpha, \bar{\boldsymbol{\vartheta}}_\alpha)$  in the frequency-parameter space, the worst-case violations are fully defined by the triple  $\{\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu, \bar{\lambda}_{\min}^\mu\}$  (immittance case) and  $\{\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu, \bar{\sigma}_{\max}^\mu\}$  (scattering case).

Let us clarify these ideas with an example. Without loss of generality, we consider a model in immittance representation, however, equivalent considerations can be drawn for the scattering case. With reference to the panels of Figure 4.1, each non-passive region  $\Theta_\alpha \in \Theta$  encloses one particular parameter value  $\boldsymbol{\vartheta}_{NP}$ , associated to models  $\mathbf{H}(s, \boldsymbol{\vartheta}_{NP})$ , whose corresponding SHH pencil shows some (finite) purely imaginary eigenvalues (see Figure 4.2,

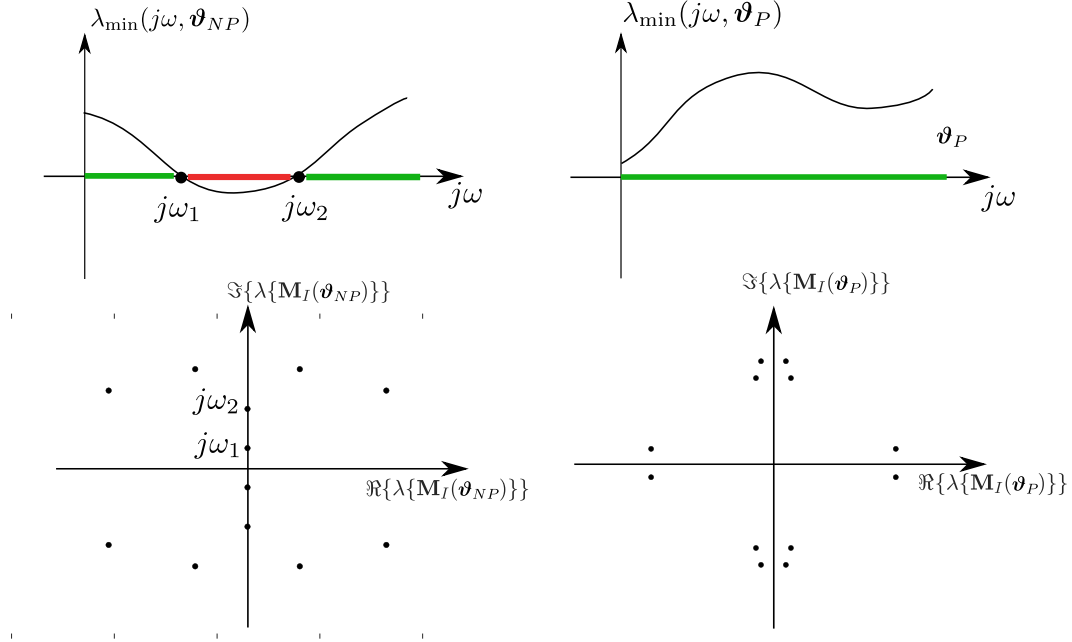


Fig. 4.2 Passivity characterization of a parameterized immittance model  $\mathbf{H}(s, \boldsymbol{\vartheta})$ . Left column: non-passive instance of  $\lambda_{\min}(j\omega, \boldsymbol{\vartheta}_{NP})$ , together with the associated SHH eigenspectrum. Right column: same for the passive parameter combination  $\boldsymbol{\vartheta}_P$ .

bottom-left panel). These eigenvalues define non-passive frequency sub-bands, as highlighted in the top-left panel of Figure 4.2. Conversely, all the parameter samples  $\boldsymbol{\vartheta}_P$  located in the green segments (left panel) or surfaces (right panel) are associated to passive models  $\mathbf{H}(s, \boldsymbol{\vartheta}_P)$ , as shown in the left column of Figure 4.2.

#### 4.2.1 Parameterized Hamiltonian eigenspectrum

Based on the parametric realizations (4.2), we generalize the definitions (3.15), and (3.16) introducing the parameterized Hamiltonian matrices  $\mathbf{M}_I(\boldsymbol{\vartheta})$  and  $\mathbf{M}_S(\boldsymbol{\vartheta})$ . The same holds for their SHH (extended and generalized) pencil extensions (see Section 3.4). In the following, we analyze how their eigenspectra varies under parameter perturbations.

Let us consider the case in which, starting from a nominal configuration  $\boldsymbol{\vartheta}_0$ , the parameter vector is perturbed as  $\boldsymbol{\vartheta} = \boldsymbol{\vartheta}_0 + \delta\boldsymbol{\vartheta}$ . Correspondingly, the SHH pencil is therefore modified, as well as its eigenspectrum. Depending on the initial spectral configuration and the perturbation extent, some initially

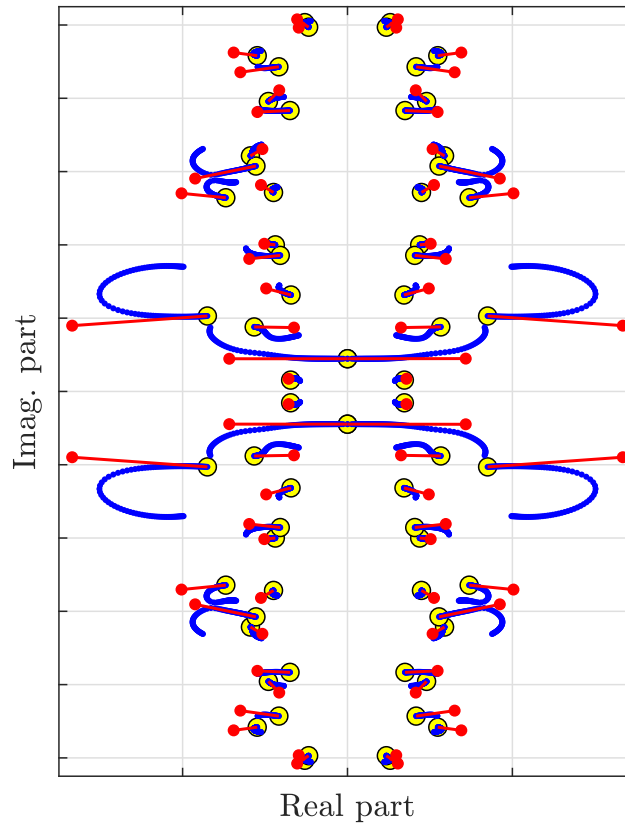


Fig. 4.3 Typical parameterized Hamiltonian eigenvalues  $\lambda_i(\boldsymbol{\vartheta}_q)$ . The big yellow dots depict the Hamiltonian (or SHH) eigenspectrum at a nominal parameter configuration  $\boldsymbol{\vartheta}_q$ , while the small blue dots represent the eigenspectrum evaluated for increasing parameter variations  $\delta\boldsymbol{\vartheta}_i$ . The red segments represent the first order perturbations  $\hat{\lambda}_i(\boldsymbol{\vartheta})$ .

non-imaginary eigenvalues may become purely imaginary, giving rise to a new passivity violation, and vice-versa. In Figure 4.3 we report a typical parameterized SHH eigenspectrum: where the yellow dots correspond to the nominal case  $\boldsymbol{\vartheta}_0$ , while the small blue dots represent the spectrum evaluated along a one-dimensional cut  $\delta\boldsymbol{\vartheta} = \mathbf{p}\delta t$ , with  $\mathbf{p} \in \mathbb{R}^\rho$ ,  $\|\mathbf{p}\| = 1$ . The passivity characterization of a parameterized model is inevitably parameter dependent.

Unfortunately, there is not a parameterized counterpart of the standard Hamiltonian method, that would enable the determination of passivity violations as a closed form algebraic problem. Looking at Figure 4.3 we see that the detection of all the passivity violations, would require a very dense sampling resolution to keep track of the highly non-linear SHH eigenvalues trajectories, that usually exhibit very high sensitivities with respect to parameter variations.

This is, however, extremely demanding in terms of CPU effort, as it requires the repeated computation of the Hamiltonian eigenvalues.

In the following, we are going to present one of the main result of this work: an adaptive strategy (originally introduced in [82]) that, based on first order perturbations of the SHH eigenvalues, enables to detect the relevant passivity violations while minimizing the number of evaluations of the SHH eigenspectrum, thus minimizing the computational effort.

Our approach is inspired by [58], that originally proposed a multi-variate passivity verification based on an adaptive parameter space sampling. Building upon the same methodology, our method sets itself as an extension and proposes a more robust solution.

### 4.3 Perturbation based passivity verification

Our multi-variate adaptive passivity check algorithm builds on the Hamiltonian approach to appropriately scan the parameter space  $\Theta$ . It can be summarized as follows:

1. In a first initialization phase, we define a reference distribution of parameter samples  $\boldsymbol{\vartheta}_q$ ,  $q = 1, \dots, \bar{q}$ , (see top-left panel of Figure 4.4);
2. for each of the available samples  $\boldsymbol{\vartheta}_q$  we compute the SHH eigenspectrum  $\Lambda_q$ ;
3. based on the spectra  $\Lambda_q$ , we determine if the tested parameter samples  $\boldsymbol{\vartheta}_q$  are passive (green dots) or not (red dots);
4. then, we determine if the set of available samples  $\boldsymbol{\vartheta}_q$  is sufficient to completely determine the passivity violations or, conversely, if additional refined are required in some regions;
5. in case a finer sampling is needed, extra  $\Delta\bar{q}$  samples are added and the process repeats from step 2) (see iterations  $J = 0, 1, 2$  in Figure 4.4 for an illustration of the adaptive sampling steps);
6. the procedure stop when no additional samples are required or a maximum number of iterations is reached (right-bottom panel of Figure 4.4).



Such an algorithm can be cast in the broad category of *exploration* and *exploitation* methods, which are standard in literature [44, 83]. We are required to infer (without actually sampling) if passivity violations do or do not exist in the continuous parameter space  $\Theta$ , relying only on a finite (and possibly small) set of “measurements”  $\boldsymbol{\vartheta}_q$ . Our multi-variate passivity verification algorithm must be therefore equipped with some predictive capabilities, to provide information on the presence (or absence) of passivity violations in the neighborhood of the points  $\boldsymbol{\vartheta}_q$ .

### 4.3.1 Derivatives of SHH pencils and eigenvalues

The predictive information is gathered in the form of first order perturbations  $\delta\lambda_i(\boldsymbol{\vartheta}_q)$  of the SHH eigenvalues, induced by small perturbations  $\delta\boldsymbol{\vartheta}$  on the parameters. Let us proceed in steps:

1. *A compact notation:*

First, let us introduce a compact notation that will prove useful in the following. Considering a generic real-valued matrix function  $\mathbf{X}(\boldsymbol{\vartheta}) \in \mathbb{R}^{m,n}$ , we collect its first-order derivatives with respect to the vector variable  $\boldsymbol{\vartheta}$  in a 3-way tensor  $\mathcal{D}\{\mathbf{X}(\boldsymbol{\vartheta})\} \in \mathbb{R}^{m,n,\rho}$ , where

$$\mathcal{D}\{\mathbf{X}(\boldsymbol{\vartheta})\}_{i,j,\nu} = \frac{\partial X_{i,j}(\boldsymbol{\vartheta})}{\partial \vartheta^\nu}. \quad (4.11)$$

The derivative tensor evaluated for the parameter combination  $\boldsymbol{\vartheta}_q$  is denoted as  $\mathcal{D}_{\boldsymbol{\vartheta}_q}\{\mathbf{X}(\boldsymbol{\vartheta})\}$ .

For each  $(i, j)$ -th element  $X_{i,j}(\boldsymbol{\vartheta})$  of  $\mathbf{X}(\boldsymbol{\vartheta})$ , the first-order Taylor expansion centered in  $\boldsymbol{\vartheta}_q$  reads

$$X_{i,j}(\boldsymbol{\vartheta}_q + \delta\boldsymbol{\vartheta}) \approx X_{i,j}(\boldsymbol{\vartheta}_q) + \nabla_{\boldsymbol{\vartheta}_q} X_{i,j}(\boldsymbol{\vartheta})^\top \cdot \delta\boldsymbol{\vartheta} \quad (4.12)$$

where  $\nabla_{\boldsymbol{\vartheta}_q}$  denotes the gradient operator evaluated at  $\boldsymbol{\vartheta}_q$ , i.e.,

$$\nabla_{\boldsymbol{\vartheta}_q} X_{i,j}(\boldsymbol{\vartheta}) = \left[ \frac{\partial X_{i,j}}{\partial \vartheta^1} \Big|_{\boldsymbol{\vartheta}^1 = \boldsymbol{\vartheta}_q^1}, \dots, \frac{\partial X_{i,j}}{\partial \vartheta^\rho} \Big|_{\boldsymbol{\vartheta}^\rho = \boldsymbol{\vartheta}_q^\rho} \right]^\top. \quad (4.13)$$

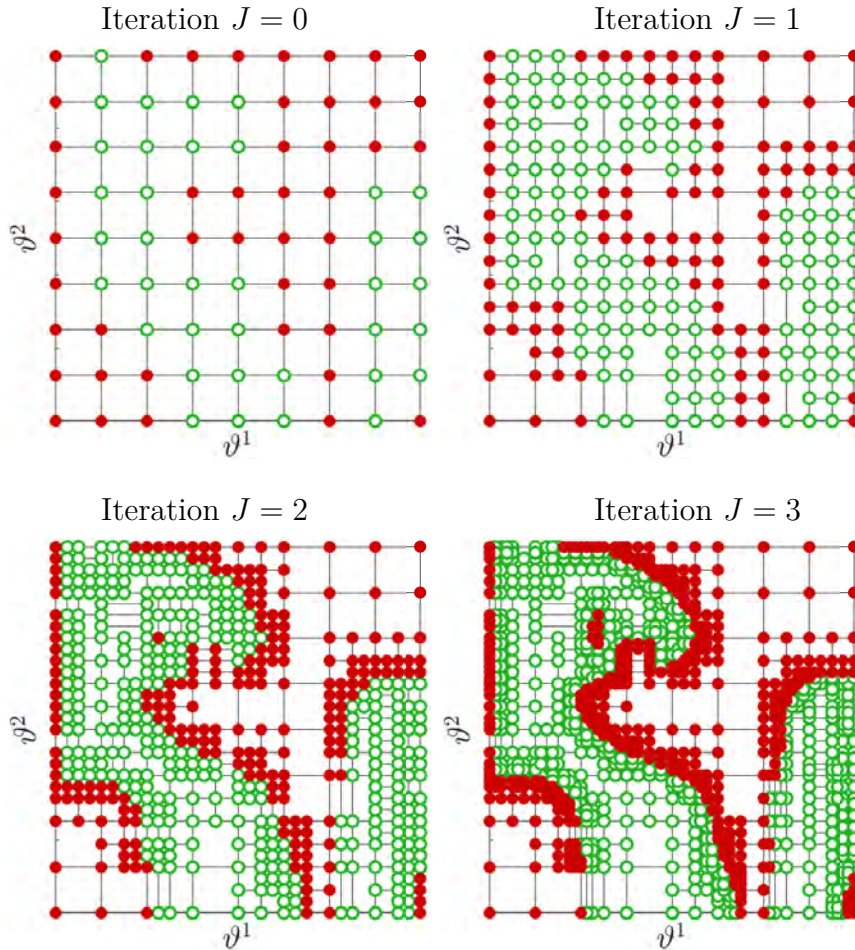


Fig. 4.4 Results of the proposed adaptive passivity verification algorithm. The four panels depict the sampled parameter space: the red and green dots correspond to non-passive and passive parameter combinations, respectively. Starting from the first coarse initialization sampling (top-left corner), the adaptive strategy iteratively refines the passivity violation boundaries and, concurrently, search for passivity violations with a small footprint (see the two bottom panels). From [82] © 2018 IEEE

Based on the above definitions, the expansion (4.12) can be compactly written for the entire matrix  $\mathbf{X}(\boldsymbol{\vartheta})$  as

$$\mathbf{X}(\boldsymbol{\vartheta}_q + \delta\boldsymbol{\vartheta}) \approx \mathbf{X}(\boldsymbol{\vartheta}_q) + \mathcal{D}_{\boldsymbol{\vartheta}_q}\{\mathbf{X}(\boldsymbol{\vartheta})\} \times_3 \delta\boldsymbol{\vartheta} \quad (4.14)$$

where  $\times_r$  denotes the  $r$ -mode tensor product (see [84] for details), so that the  $(i, j)$ -th vector element of  $\mathcal{D}_{\boldsymbol{\vartheta}_q}\{\mathbf{X}(\boldsymbol{\vartheta})\} \times_3 \delta\boldsymbol{\vartheta}$  corresponds to the vector inner product  $\nabla_{\boldsymbol{\vartheta}_q} X_{i,j}(\boldsymbol{\vartheta})^\top \cdot \delta\boldsymbol{\vartheta}$ .

2. *Derivatives of Hamiltonian matrices:*

Then, we derive a first-order approximation for the perturbed SHH pencil  $(\mathbf{M}(\boldsymbol{\vartheta}_q + \delta\boldsymbol{\vartheta}), \mathbf{Y})$  where, dropping the subscripts  $(I, S)$ , we concurrently consider immittance and scattering representations. The first step is to compute the derivatives (with respect to  $\boldsymbol{\vartheta}$ ) of the realization matrices. To this end, let us consider a descriptor realization for  $\mathbf{H}(s, \boldsymbol{\vartheta})$ , for which the parameterized realization matrices  $\mathbf{A}(\boldsymbol{\vartheta})$  and  $\mathbf{C}(\boldsymbol{\vartheta})$  can be expressed as linear combinations of the basis functions, see (2.86).

Adopting the above notation, we can easily define the derivatives of the realization matrices, as

$$\mathcal{D}\{\mathbf{A}(\boldsymbol{\vartheta})\} = \sum_{\ell \in \mathcal{I}} \mathbf{A}_\ell \circ \nabla \xi_\ell(\boldsymbol{\vartheta}), \quad (4.15)$$

and, similarly, for  $\mathbf{C}(\boldsymbol{\vartheta})$ . The operator  $\circ$  denotes the outer product [84], that stores the individual products of  $\mathbf{A}_\ell$  with the element of  $\nabla \xi_\ell(\boldsymbol{\vartheta})$  along the third dimension of the tensor. Based on the definition (3.22), (3.23), these expressions are used to construct the derivative tensor  $\mathcal{D}\{\mathbf{M}(\boldsymbol{\vartheta})\}$  of the Hamiltonian matrix  $\mathbf{M}(\boldsymbol{\vartheta})$ , used to compute its first-order perturbation through (4.14), as

$$\mathbf{M}(\boldsymbol{\vartheta}_q + \delta\boldsymbol{\vartheta}) \approx \mathbf{M}(\boldsymbol{\vartheta}_q) + \mathcal{D}_{\boldsymbol{\vartheta}_q}\{\mathbf{M}(\boldsymbol{\vartheta})\} \times_3 \delta\boldsymbol{\vartheta}. \quad (4.16)$$

The perturbed SHH pencil thus reads  $(\mathbf{M}(\boldsymbol{\vartheta}_q + \delta\boldsymbol{\vartheta}), \mathbf{Y})$ .

3. *First-order approximation of SHH eigenvalues:*

Let us consider a (finite) eigenvalue  $\lambda_i(\boldsymbol{\vartheta}_q)$  of  $(\mathbf{M}(\boldsymbol{\vartheta}_q), \mathbf{Y})$ , and let us denote  $\mathbf{u}_i, \mathbf{v}_i$  the associated right and left eigenvectors. Standard results in eigenvalue perturbation theory [85] enables us to derive a first-order

approximation for the trajectory of  $\lambda_i(\boldsymbol{\vartheta})$ , when subject to a perturbation  $\boldsymbol{\vartheta} = \boldsymbol{\vartheta}_q + \delta\boldsymbol{\vartheta}$ , as

$$\hat{\lambda}_i(\boldsymbol{\vartheta}) \approx \lambda_i(\boldsymbol{\vartheta}_q) + \delta\lambda_i(\boldsymbol{\vartheta}_q), \quad (4.17)$$

where the perturbation term reads

$$\delta\lambda_i(\boldsymbol{\vartheta}_q) \approx \frac{\mathcal{D}_{\boldsymbol{\vartheta}_q}\{\mathbf{M}(\boldsymbol{\vartheta})\} \times_1 \mathbf{v}_i^* \times_2 \mathbf{u}_i \times_3 \delta\boldsymbol{\vartheta}}{\mathbf{v}_i^H \cdot \mathbf{Y} \cdot \mathbf{u}_i} \quad (4.18)$$

In Figure 4.3 the first order eigenvalues perturbations are depicted with solid red lines that, as expected, are tangent to the exact non-linear eigenvalues trajectories at  $\boldsymbol{\vartheta}_q$ . Not surprisingly, the linear predictions (represented with the terminal red dots) are valid only for small  $\delta t$ .

We remark that the “length” of the linear perturbations (more in details, their 2-norm) provides useful information associated to the sensitivity of the eigenvalues to parameter variations. As we will see in Section 4.3.4, this is one of the key points for the improved reliability of this approach.

### 4.3.2 Adaptive sampling via Hamiltonian spectral perturbation

The linear perturbation scheme we just introduced enables to make a prediction on the SHH eigenspectrum in correspondence to a small neighborhood of a given parameter combination  $\boldsymbol{\vartheta}_q$ . This capability is exploited by our passivity verification scheme to infer the onset of possible violations in unexplored regions of the parameter space. In terms of computational efficiency, this is highly beneficial. Indeed, computing the linear predictions through (4.17) comes almost at a zero-cost and can be repeatedly executed to fully explore the SHH eigenspectrum in the neighborhood of  $\boldsymbol{\vartheta}_q$ ; the exact SHH eigenvalues are evaluated only in case some violations are predicted, dramatically reducing the overall computational cost.

Our multi-variate adaptive algorithm consists of two main steps: an initial *grid initialization* and a subsequent adaptive *grid refinement*.

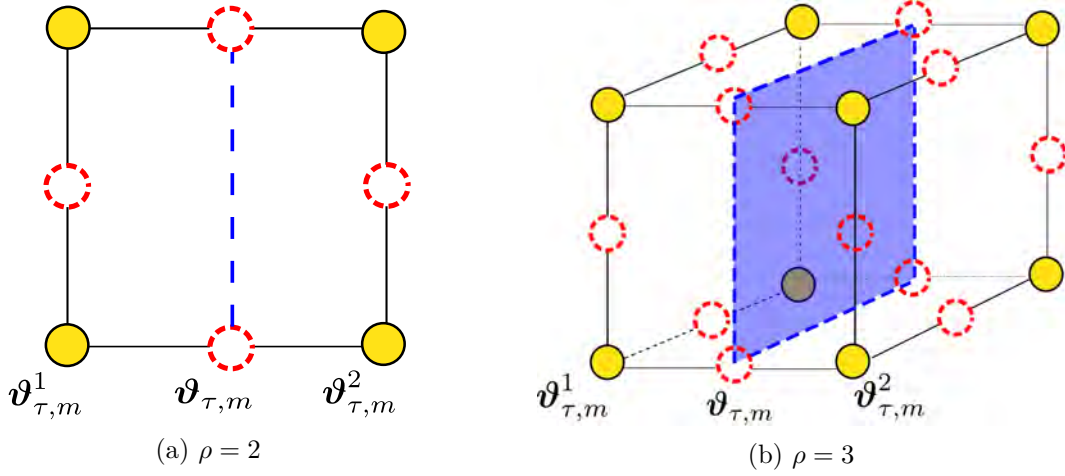


Fig. 4.5 Elementary cell  $\Gamma_\tau$  in 2D (left) and 3D (right) being refined along the first direction  $\nu = 1$ . Filled yellow dots and red dashed dots represent vertices and edge midpoints, respectively. From [82] © 2018 IEEE

### Grid initialization

As a first step, we partition the parameter space  $\Theta$  according to a (hyper)-cubic Cartesian lattice. Assuming  $j_0^\nu$  to be the initial refinement level associated with the  $\nu$ -th dimension, this is achieved subdividing each parameter space direction  $\nu = 1, \dots, \rho$  in  $2^{j_0^\nu}$  in equally sized sub-intervals, whose end-points are collected in set  $\mathcal{S}_{j_0^\nu}$ , defined as

$$\mathcal{S}_{j_0^\nu} = \{k 2^{-j_0^\nu}, k = 0, \dots, 2^{j_0^\nu}\} \quad (4.19)$$

The initial set of vertices  $\mathbf{v}_q$ , uniquely identified with a global index  $q = 1, \dots, \bar{q}$ , are collected in set  $\mathcal{V}_0$ , defined by the Cartesian product

$$\mathcal{V}_0 = \mathcal{S}_{j_0^1} \times \mathcal{S}_{j_0^2} \times \dots \times \mathcal{S}_{j_0^\rho} \quad (4.20)$$

The above partitioning is completely determined by the triple  $\{\mathcal{V}_0, \mathcal{E}_0, \mathcal{F}_0\}$ , where  $\mathcal{F}_0$  collects the elementary hyper-cubic cells  $\Gamma_\tau$ ,  $\tau = 1, \dots, \bar{\tau}$ , while  $\mathcal{E}_0$  collects the cell's edges connecting two adjacent vertices. With reference to the elementary cell  $\Gamma_\tau$  (depicted in Figure 4.5 for the bi- and tri-parameter cases), we denote the associated edges as  $e_{\tau,m} \in \mathcal{E}_0$ ,  $m = 1, \dots, \rho 2^{\rho-1}$  and their endpoint vertices as  $\mathbf{v}_{m,\tau}^\gamma \in \mathcal{V}_0$ ,  $\gamma = 1, 2$

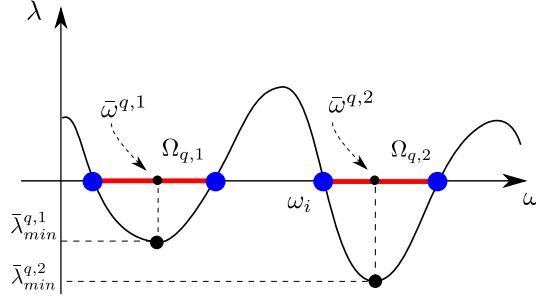


Fig. 4.6 Representation of one eigenvalue of  $\Psi_I(j\omega, \boldsymbol{\vartheta}_q)$  corresponding to a fixed parameter value  $\boldsymbol{\vartheta}_q$ . The eigenvalue  $\lambda$  (solid black line) crosses the zero baseline for  $\omega = \omega_i$  (the imaginary SHH eigenvalues, blue large dots), delimiting non-passive frequency bands  $\Omega_{q,1}, \Omega_{q,2}$  (highlighted as thick red lines). Local minima are denoted as small black dots. From [82] © 2018 IEEE

As the initial grid is available, we compute the SHH eigenvalues  $\lambda_{q,i} = \lambda_i(\boldsymbol{\vartheta}_q)$  in correspondence of each vertex  $\boldsymbol{\vartheta}_q \in \mathcal{V}_0$ , together with the associated left and right eigenvectors  $\mathbf{v}_{q,i}, \mathbf{u}_{q,i}$ ; we collect the triples  $\{\lambda_{q,i}, \mathbf{v}_{q,i}, \mathbf{u}_{q,i}\}$  in set  $\mathcal{L}_0$ . Then, considering only the purely imaginary eigenvalues  $\lambda_{q,i} = j\omega_{q,i}$ , we extract the corresponding frequencies  $\omega_{q,i}$  that we collect in set  $\Omega_q$ . On the basis of Theorem 3, if  $\Omega_q = \emptyset$  we flag the vertex  $\boldsymbol{\vartheta}_q$  as passive (identified with green dots in Figure 4.4); otherwise,  $\boldsymbol{\vartheta}_q$  is flagged as non-passive (identified as red filled dots in Figure 4.4). The non-passive frequency sub-bands delimited by the frequency values  $\omega_{q,i} \in \Omega_q$  are identified as  $\Omega_{q,\beta}$ , with  $\beta = 1, \dots, \bar{\beta}_q$ . Finally, for each non-passive sub-band  $\Omega_{q,\beta}$  we determine the worst-case passivity violation  $\bar{\lambda}_{\min}, \bar{\sigma}_{\max}$  occurring at frequency  $\bar{\omega}_\mu = \bar{\omega}^{q,\beta}$ . See Figure 4.6 for an illustration.

Together with the parameter combination  $\bar{\boldsymbol{\vartheta}}_\mu = \boldsymbol{\vartheta}_q$ , we collect the worst-case violation data  $\{\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu, \bar{\lambda}_{\min}^\mu\}$  (immittance case) and  $\{\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu, \bar{\sigma}_{\max}^\mu\}$  (scattering case), in a common set  $\mathcal{W}_\mu, \mu = 1, \dots, \bar{\mu}$ , updated anytime other violations are detected.

### Grid refinement

Based on the initial coarse sampling, the location estimates of passivity violations (if any) are iteratively refined in this second iterative step. At any iteration  $J = 1, 2, \dots$ , we are given with sets  $\mathcal{F}_{J-1}, \mathcal{V}_{J-1}, \mathcal{E}_{J-1}$ , describing the structure of the adaptive grid at the previous iteration, together with  $\mathcal{L}_{J-1}$  that stores the eigenvalues/eigenvectors evaluated at each grid node.

This information is employed to process all the elementary cells  $\Gamma_\tau$ , looping over the elements of set  $\mathcal{F}_{J-1}$ . First, in regard to the generic cell  $\Gamma_\tau$ , we define the midpoints  $\boldsymbol{\vartheta}_{\tau,m}$  associated with the edges  $e_{\tau,m}$  connecting the vertices  $\boldsymbol{\vartheta}_{\tau,m}^\gamma$ ,  $\gamma = 1, 2$  (see Figure 4.5 for an illustration). Then, considering the  $m$ -th edge  $e_{\tau,m}$ , the eigenvalues  $\lambda_{\tau,m;i}^\gamma$  at the reference vertices  $\boldsymbol{\vartheta}_{\tau,m}^\gamma$  are perturbed through (4.17) in order to evaluate their predictions  $\hat{\lambda}_{\tau,m;i}^\gamma$  in correspondence to the midpoint  $\boldsymbol{\vartheta}_{\tau,m}$ .

**Remark 10.** *Due to the 4-quadrant symmetry of the SHH eigenspectrum we consider only the finite eigenvalues in the 2<sup>nd</sup> quadrant.*

Based on these estimates, we infer the presence (or absence) of localized passivity violations along the edge  $e_{\tau,m}$ ; anytime the following conditions hold

1.  $\Re\{\hat{\lambda}_{\tau,m;i}^\gamma - \lambda_{\tau,m;i}^\gamma\} > 0$  for some  $\gamma$ , implying that  $\lambda_{\tau,m;i}^\gamma$  is moving closer to the imaginary axis, and
2.  $\Re\{\hat{\lambda}_{\tau,m;i}^\gamma\} > -\delta$  for the same  $\gamma$  fulfilling above condition, where  $\delta > 0$  is a small predefined threshold,

the edge is refined adding the *critical* midpoint  $\boldsymbol{\vartheta}_{\tau,m}$  as a new vertex. In order to subdivide the cell  $\Gamma_\tau$  in structured partitions, the detection of critical midpoints triggers the addition of other  $2^{p-1}$  midpoints that complete the partitioning of the current cell in two smaller sub-cells. Whenever a new sub-cell is created, it is added to the set  $\mathcal{F}_J$ , while its vertices and edges are added to sets  $\mathcal{V}_J$  and  $\mathcal{E}_J$ , respectively. Finally, we flag the new vertices as passive or non-passive by evaluating the SHH eigenvalues that are stored, together with the eigenvectors, in set  $\mathcal{L}_J$ .

A graphical illustration of the above refinement rules is provided in Figure 4.7: only the first case highlights the presence of a critical midpoint, since the considered eigenvalue is moving closer to the imaginary axis. Conversely, cases 2) and 3) are considered *not critical* since the eigenvalue predictions are either sufficiently far from the imaginary axis or, indeed, reveal the tendency of the considered eigenvalues to move in the opposite direction.

The above procedure is performed only in case both the edge endpoints are passive, since small localized passivity violations may be hidden between

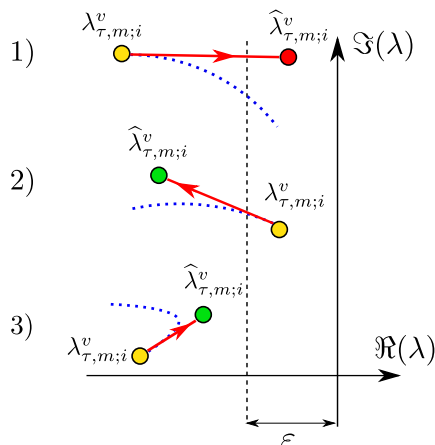


Fig. 4.7 Graphical representation of the adaptive refinement rules. Yellow dots: SHH eigenvalues  $\lambda_{\tau,m;i}^v$  computed at vertex  $\vartheta_{\tau,m}$ ; green and red dots: linear prediction (red thick lines with arrows) of perturbed eigenvalues  $\hat{\lambda}_{\tau,m;i}^v$  at the edge midpoint  $\vartheta_{\tau,m}$ ; blue dotted lines: exact eigenvalue trajectories along the edge  $\mathbf{e}_{\tau,m}$ . From [82] © 2018 IEEE

the two nodes. In other circumstances, however, we can already deduce the presence of passivity violations without any further information, in particular:

- it both end-points are non-passive: in this occurrence no refinement is required, as a local violation has been already detected;
- if one endpoint is non-passive and the other is passive: we already detected the presence of a passivity violation; the edge needs further refinements to accurately detect the violation boundaries, located between the considered edges.

The adaptive refinement procedure is equipped with a safe termination threshold, achieved when the number of iterations  $J$  reaches a maximum level  $J_{\max}$ . This is particularly useful in the case of nearly lossless models, characterized by SHH eigenvalues that are uniformly close to the imaginary axis and, depending on the selected threshold  $\delta$ , may trigger unnecessary grid refinements. Such a termination rule sets an upper bound on the sampling resolution, potentially undermining the reliability since small passivity violations may be missed. This occurrence can be circumvented by setting  $J_{\max}$  to a sufficiently large value; in our tests,  $J_{\max}$  in the range  $[5, 10]$  proved to be a reasonable trade-off between computational complexity and sampling



resolution. In case residual passivity violations remain undetected, these would be practically unimportant.

### 4.3.3 A relevant test-case

The adaptive capabilities of the presented passivity verification algorithm are reported in the four panels of Figure 4.4, showing the sampled parameter space over successive refinement adaptive steps. The green and red filled dots denote, respectively, *passive* and *non-passive* parameter instances.

We start with a  $9 \times 9$  uniform grid (obtained setting  $j_0^1 = j_0^2 = 3$ ) at the first initialization iteration ( $J = 0$ ); although some initial estimates on the distribution of passive/non-passive regions can already be inferred, their resolution is very poor. As the iterations proceed, the proposed adaptive algorithm better tracks the boundaries of violation regions until, at iteration  $J = 3$ , they are perfectly resolved.

This example is particularly relevant, as it demonstrates that:

- the proposed adaptive strategy samples the parameter space only where strictly required; no unnecessary samples are added in uniformly passive or active regions;
- the predictive properties that we embedded through the linear perturbation of the SHH eigenvalues guarantee an improved reliability. In fact, despite the small violation located in the top-left corner remained undetected in the first two iterations (in a seemingly uniformly passive region), the algorithm “predicted” its presence and was able to identify it at iteration  $J = 2$ .

For this particular case, the absence of other hidden passivity violations has been confirmed through a verification test based on a fine gridding.

### 4.3.4 Comparison with other approaches

We now compare the above perturbation-based adaptive sampling with the method presented in [58]. The first major improvement of our method consists

in the extension to higher-dimensional spaces (2-D, 3-D), whereas the method in [58] is limited to one independent parameter only.

Further, the proposed approach highlights and solves some reliability issues that affect [58]. The reason for this, is better understood recalling that the adaptive procedure in [58] relies on the *spectral distance* function  $\psi(\vartheta)$

$$\psi(\vartheta) = \min_{\lambda(\vartheta) \in \Lambda(\vartheta)} \frac{|\Re\{\lambda(\vartheta)\}|}{\varrho(\vartheta)} \quad (4.21)$$

with  $\varrho(\vartheta)$  the spectral radius of  $\Lambda_g$ ; this function turns out to be particularly useful, as it measures the (normalized) distance between the imaginary axis and the closest eigenvalue. Parameter occurrences at which  $\psi(\vartheta) = 0$  highlight the presence of purely imaginary SHH eigenvalues, denoting the presence of non-passive behaviors; otherwise, if  $\psi(\vartheta) \neq 0$ , there can not be purely imaginary eigenvalues, which implies local passivity.

Although the function  $\psi(\vartheta)$  is non-smooth by definition, the approach devised in [58] considers its abrupt variations as indicative of possible passivity violations. Hence, the adaptive rule proposed in [58] can be summarized as follows: if the degree of smoothness of  $\psi(\vartheta)$  is higher than a predefined threshold, no actions are required; otherwise, a refinement procedure is triggered to better explore the parameter space.

In this regard, in [58] it has been pointed out that sharp and localized passivity violations can be detected by increasing the number of refinement iterations; this is however not true. Indeed, in the left panel of Figure 4.8 we depict the function  $\psi(\vartheta)$  obtained with the method [58] while, on the right panel, the one obtained adopting the proposed strategy (computed with a subsequent reconstruction and not used for actual calculations). The onset of the small violation at  $\vartheta \approx 28$  has been completely missed by the approach [58], while it is detected and accurately tracked by the proposed method.

The reason behind the improved accuracy becomes evident looking at Figure 4.9, where we depict the trajectories of the two SHH eigenvalues that are closest to the imaginary axis, in correspondence to a parameter range  $[\vartheta_1, \vartheta_2]$  that encloses the sharp violation. The algorithm [58] detects only the presence of the eigenvalue  $\lambda_1(\vartheta)$  (since closer to the imaginary axis in correspondence to

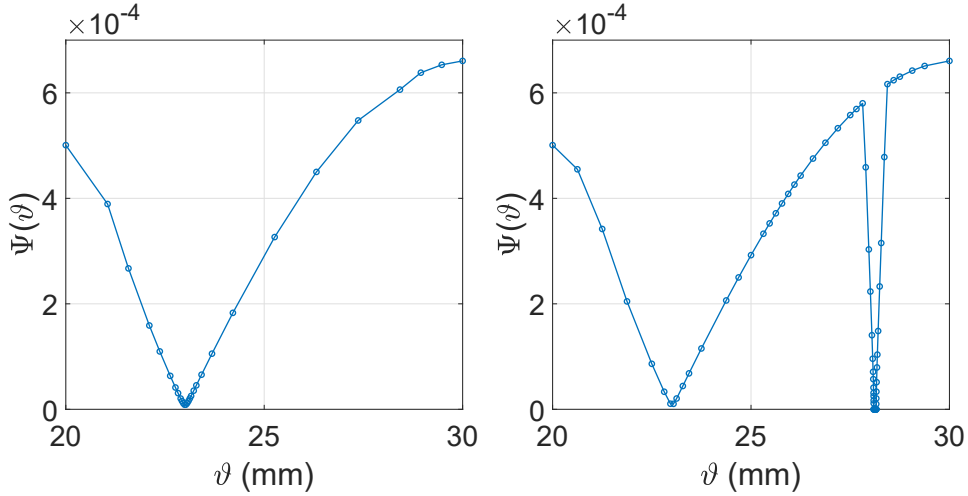


Fig. 4.8 Comparing the spectral distance functions for the two approaches. Left panel: function  $\psi(\vartheta)$  adaptively sampled with the bisection method [58]. Right panel: reconstruction of the sampled function  $\psi(\vartheta)$  resulting from the proposed approach. From [82] © 2018 IEEE

the reference values  $\vartheta = \vartheta_1$  and  $\vartheta = \vartheta_2$ ) and misses the “fast” eigenvalue  $\lambda_2(\vartheta)$  that undergoes a fast passive / non-passive / passive transition.

Conversely, the additional information about the SHH eigenvalues sensitivity carried by their parameter-derivatives enables to detect the “critical speed” of the eigenvalue  $\lambda_2(\vartheta)$ , triggering a refinement that, in turn, detects the presence of a violation.

## 4.4 Enforcing passivity of parameterized models

Once a complete description of the passivity violation areas is obtained, we can proceed with the subsequent *passivity enforcement* step, in which all the detected violations are compensated. Following [58], we consider a parameterized extension of the *local perturbation* scheme, outlined in Section 3.5.3 for the non-parametric case. Hence, the need of a reliable passivity verification algorithm becomes justified, since undetected violations can not be compensated.

We highlight that [86] has recently presented a parameterized extension of the LMI approach (discussed in Section 3.5.1 for the univariate setting) that,

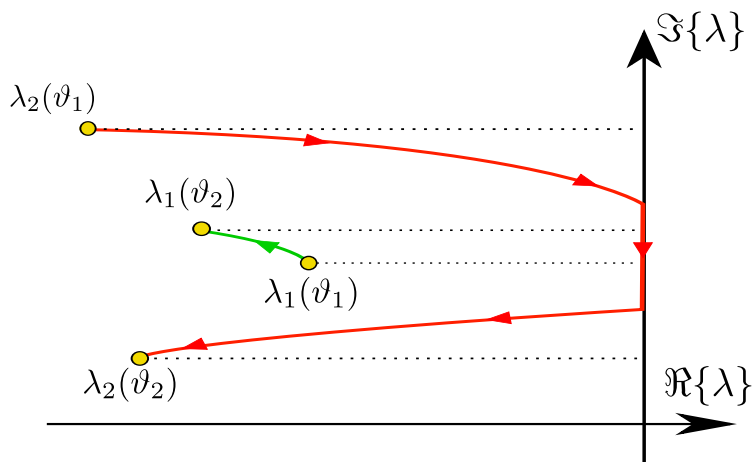


Fig. 4.9 Hamiltonian eigenvalues trajectories associated to the spectral distance function of Figure 4.8. The red eigenvalue corresponds to the “fast” eigenvalue, while the green trajectory corresponds to the “slow” eigenvalue. From [82] © 2018 IEEE

although theoretically guaranteed to generate a uniformly passive model, it is feasible only for small-scale problems. Instead, the adopted perturbation approach has a broader range of applicability, at the cost of a required iterative process.

The parametric passivity enforcement procedure is similar to its univariate counterpart: starting from an originally non-passive model  $\mathbf{H}(s, \boldsymbol{\vartheta})$ , we need to find a suitably small perturbation  $\Delta\mathbf{H}(s, \boldsymbol{\vartheta})$  such that the perturbed model

$$\widehat{\mathbf{H}}(s, \boldsymbol{\vartheta}) = \mathbf{H}(s, \boldsymbol{\vartheta}) + \Delta\mathbf{H}(s, \boldsymbol{\vartheta}) \quad (4.22)$$

is uniformly passive in the frequency-parameter space  $j\mathbb{R} \times \Theta$ , remaining concurrently accurate on the band of interest. Based on the considerations of Section 3.5, expressing  $\Delta\mathbf{H}(s, \boldsymbol{\vartheta})$  in the following descriptor form proves to be particularly suitable

$$\Delta\mathbf{H}(s, \boldsymbol{\vartheta}) = \Delta\mathbf{C}(\boldsymbol{\vartheta}) (s\mathbf{E} - \mathbf{A}(\boldsymbol{\vartheta}))^{-1} \mathbf{B} \quad (4.23)$$

where only the state-output matrix  $\Delta\mathbf{C}(\boldsymbol{\vartheta})$  is perturbed. Recalling that, in the descriptor form, we are allowed to expand the realization matrices as linear combinations of the parameter basis functions (2.86), we define the unknown

perturbation matrix  $\Delta\mathbf{C}(\boldsymbol{\vartheta})$  as

$$\Delta\mathbf{C}(\boldsymbol{\vartheta}) = \sum_{\ell \in \mathcal{I}_N} \Delta\mathbf{C}_\ell \xi_\ell(\boldsymbol{\vartheta}), \quad (4.24)$$

where  $\Delta\mathbf{C}_\ell$  are the optimization variables that we collect in a block matrix as  $\mathbf{X} = (\Delta\mathbf{C}_1, \dots, \Delta\mathbf{C}_\ell)$ .

For the further developments, it is convenient to write (4.23) in a more compact form, where the decision variables  $\mathbf{X}$  appear explicitly. To this end, defining the auxiliary function  $\mathbf{Z}(s, \boldsymbol{\vartheta})$  as

$$\mathbf{Z}(s, \boldsymbol{\vartheta}) = \left\{ [\xi_1(\boldsymbol{\vartheta}), \dots, \xi_{\bar{\ell}}(\boldsymbol{\vartheta})]^\top \otimes \mathbf{I}_{P(\bar{n}+1)} \right\} [s\mathbf{E} - \mathbf{A}(\boldsymbol{\vartheta})]^{-1} \mathbf{B} \quad (4.25)$$

it is easy to prove that the unknown perturbation  $\Delta\mathbf{H}(s, \boldsymbol{\vartheta})$  can be written as

$$\Delta\mathbf{H}(s, \boldsymbol{\vartheta}) = \mathbf{X} \cdot \mathbf{Z}(s, \boldsymbol{\vartheta}), \quad (4.26)$$

In the end, enforcing passivity of parameterized models requires solving the problem

$$\begin{aligned} \min \quad & \|\Delta\mathbf{H}(s, \boldsymbol{\vartheta})\| \\ \text{s.t.} \quad & \widehat{\mathbf{H}}(s, \boldsymbol{\vartheta}) \text{ is passive} \end{aligned} \quad (4.27)$$

that, while minimizing the model perturbations  $\Delta\mathbf{C}_\ell$ , removes the passivity violations through suitable constraints, that will be discussed in the following.

#### 4.4.1 Local linearized passivity constraints

Based on the information collected in set  $\mathcal{W}_\mu$  during the passivity verification step, we formulate a set of (linearized) passivity constraints. This is enabled by exploiting the information about the worst-case passivity violations  $\mathcal{W}_\mu$ , gathered during the passivity check. The passivity constraints must be adequately defined according to the selected model representation (immittance or scattering); let us analyze the two cases separately.

**Immittance:**

Let us consider the worst-case passivity violation  $\mathcal{W}_\mu$  that, in the immittance case, is defined by the triple  $\{\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu, \bar{\lambda}_{\min}^\mu\}$ . In order to construct a local passivity constraint, we must perturb the negative eigenvalue  $\bar{\lambda}_{\min}^\mu$  of  $\mathbf{H}(\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu) + \mathbf{H}^H(\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu)$  to enforce its local non-negativity, as

$$\bar{\lambda}_{\min}^\mu + \Delta\bar{\lambda}_{\min}^\mu \geq 0 \quad (4.28)$$

by means of a suitable model perturbation  $\Delta\mathbf{H}(s, \boldsymbol{\vartheta})$ . The local passivity constraint associated with the  $\mu$ -th sampled violation is thus formulated as

$$-\Delta\bar{\lambda}_{\min}^\mu \leq \bar{\lambda}_{\min}^\mu. \quad (4.29)$$

Following simple eigenvalue perturbation arguments [85], we can explicitly write the first-order approximation for  $\Delta\bar{\lambda}_{\min}^\mu$  induced by the perturbation  $\Delta\mathbf{H}(\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu)$ , that reads

$$\Delta\bar{\lambda}_{\min}^\mu \approx 2\Re\{\mathbf{v}_\mu^H \Delta\mathbf{H}(\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu) \mathbf{v}_\mu\}, \quad (4.30)$$

where  $\mathbf{v}_\mu$  is the right eigenvector of  $\mathbf{H}(\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu)$  associated with  $\bar{\lambda}_{\min}^\mu$ . Plugging the approximation (4.30) in (4.29) leads to the following *linearized* passivity constraint

$$-2\Re\{\mathbf{v}_\mu^H \Delta\mathbf{H}(\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu) \mathbf{v}_\mu\} \leq \bar{\lambda}_{\min}^\mu. \quad (4.31)$$

Finally, exploiting the definition of  $\Delta\mathbf{H}(s, \boldsymbol{\vartheta})$  in (4.26), through simple algebraic manipulations the linear constraint (4.31) can be re-written as

$$\underbrace{-2\Re\{\mathbf{Z}(\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu) \mathbf{v}_\mu\}^\top \otimes \mathbf{v}_\mu^H}_{\mathbf{y}_\mu^\top} \cdot \mathbf{x} \leq \bar{\lambda}_{\min}^\mu. \quad (4.32)$$

where decision variables  $\mathbf{x} = \text{vet}\{\mathbf{X}\}$  are explicit. Equation 4.32 forms the local passivity constraint to be enforced for all the passivity violations in  $\mathcal{W}$ .

**Scattering:**

In the scattering case, the  $\mu$ -th worst-case passivity violation is defined by the triple  $\{\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu, \bar{\sigma}_{\max}^\mu\}$ . Similarly to the immittance case, local passivity is enforced by suitably perturbing the non-passive singular value  $\bar{\sigma}_{\max}^\mu$ . In details, we aim for a local perturbation  $\Delta\bar{\sigma}_{\max}^\mu$  such that the perturbed singular value  $\bar{\sigma}_{\max}^\mu + \Delta\bar{\sigma}_{\max}^\mu$  satisfies the passivity condition

$$\Delta\bar{\sigma}_{\max}^\mu \leq 1 - \bar{\sigma}_{\max}^\mu \quad (4.33)$$

The model perturbation  $\Delta\mathbf{H}(\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu)$  induces a variation on the singular values that, at the first order, can be approximated as

$$\Delta\bar{\sigma}_{\max}^\mu \approx \Re\{\mathbf{u}_\mu^H \Delta\mathbf{H}(j\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu) \mathbf{v}_\mu\} \quad (4.34)$$

Expressing the above equation in terms of the decision variable  $\mathbf{x}$  through (4.26), we obtain the following linearized passivity constraint

$$\underbrace{\Re\{[\mathbf{Z}(j\bar{\omega}_\mu, \bar{\boldsymbol{\vartheta}}_\mu) \mathbf{v}_\mu]^\top \otimes \mathbf{u}_\mu^H\}}_{\mathbf{z}_\mu^\top} \cdot \mathbf{x} \leq 1 - \bar{\sigma}_{\max}^\mu \quad (4.35)$$

to be enforced for any violation in  $\mathcal{W}_\mu$ .

**A unified notation**

In both the *immittance* and *scattering* representation, the passivity constraints come in the form of linear inequalities. If we define now

$$\begin{aligned} \mathbf{g}_\mu &= -\mathbf{y}_\mu, & h_\mu &= \bar{\lambda}_{\min}^\mu & \text{immittance,} \\ \mathbf{g}_\mu &= \mathbf{z}_\mu, & h_\mu &= 1 - \bar{\sigma}_{\max}^\mu & \text{scattering,} \end{aligned}$$

the above passivity constraints can be unified under the common notation

$$\mathbf{g}_\mu^\top \cdot \mathbf{x} - h_\mu \leq 0, \quad \mu = 1, \dots, \bar{\mu}. \quad (4.36)$$

### 4.4.2 Preserving the accuracy

While enforcing the above passivity constraints, the model accuracy is preserved by minimizing the model perturbation  $\|\Delta\mathbf{H}(s_\kappa, \boldsymbol{\vartheta}_\kappa)\|$  over a set of predefined frequency-parameter combinations  $(s_\kappa, \boldsymbol{\vartheta}_\kappa)$ ,  $\kappa = 1, \dots, \bar{\kappa}$ ; these, for instance, can be selected as belonging to the dataset used to construct the original model. This requirement is readily translated in the form of a cost functional, defined as

$$F(\mathbf{x}) = \sum_{\kappa=1}^{\bar{\kappa}} \|\Delta\mathbf{H}(s_\kappa, \boldsymbol{\vartheta}_\kappa)\|_F^2 = \sum_{\kappa=1}^{\bar{\kappa}} \|\mathbf{X} \cdot \mathbf{Z}(s_\kappa, \boldsymbol{\vartheta}_\kappa)\|_F^2 \quad (4.37)$$

**Remark 11.** *Depending on the problem at hand, suitable frequency/parameter weighting schemes can be applied with minor modifications to the cost functional  $F(\mathbf{x})$ .*

### 4.4.3 Passivity enforcement iteration

Based on the cost-function (4.37) and the constraints (4.36), we are ready to formulate the passivity enforcement problem as

$$\begin{aligned} \min_{\mathbf{x}} F(\mathbf{x}) \\ \text{s.t. } \mathbf{g}_\mu^\top \cdot \mathbf{x} - h_\mu \leq 0, \quad \mu = 1, \dots, \bar{\mu} \end{aligned} \quad (4.38)$$

As in the univariate setting, the linearization of the real eigenvalues/singular values perturbations lead the constraints (4.36) to be valid only up to a first-order approximation. Thus, the solution of (4.38) is not guaranteed to return a uniformly passive model. Following the univariate scheme, this problem is addressed repeatedly solving (4.38), until all the violations are eliminated (see later Figure 4.13 for an illustration).

**Remark 12.** *As detailed in Section 3.5.3, compensating a discrete set of passivity violations by means of the perturbation  $\Delta\mathbf{H}(s, \boldsymbol{\vartheta})$ , may induce the onset of new violations in already passive regions. Based on a parametric extension of [81], this problem can be solved by formulating a more robust passivity enforcement scheme that prevents the rise of new unwanted violations.*



In Algorithm (1) we provide a high-level pseudo-code for the proposed passivity enforcement scheme, where a distinction between the passivity enforcement iteration (outer iterations) and passivity assessment iterations (inner iterations) can be appreciated.

---

**Algorithm 1** Multi-variate passivity enforcement
 

---

**Require:** Initial (possibly) non-passive parameterized model (2.64)

```

1: repeat (passivity enforcement iterations)
2:   Compute the SHH eigenvalues at initial grid points in  $\mathcal{V}_0$  (Sec. 4.3.2)
3:   Collect the eigenvalues/eigenvectors in set  $\mathcal{L}_0$  (Sec. 4.3.2)
4:   Initialize local passivity violations  $\mathcal{W}$  (Sec. 4.3.2)
5:   for  $J = 1, \dots, J_{\max}$  (passivity check iterations) do
6:     Loop over the cells  $\Gamma_\tau$  and determine which must be refined
7:     if some cells must be refined then
8:       Refine critical cells (Sec. 4.3.2)
9:       Compute (and collect in  $\mathcal{L}_J$ ) the SHH eigs at new points
10:      Update set  $\mathcal{W}$ 
11:     else
12:       break
13:     end if
14:   end for
15:   if  $\mathcal{W} \neq \emptyset$  (model is not passive) then
16:     Formulate local passivity constraints and solve (4.38)
17:     Apply the perturbation  $\Delta\mathbf{H}(s, \boldsymbol{\vartheta})$  to the current model as in (4.22).
18:   end if
19: until  $\mathcal{W} = \emptyset$  (model is passive)
20: return Passive model  $\widehat{\mathbf{H}}(s, \boldsymbol{\vartheta})$ 

```

---

## 4.5 Numerical experiments

In this section, we will test the performances of the proposed parameterized passivity verification and enforcement strategies. Our test-bench is composed of EM structures, parameterized by up to 3 parameters. The experiments have been performed on a standard 16 GB of RAM laptop, running an Intel CPU (Core-i7 2.3 GHz).

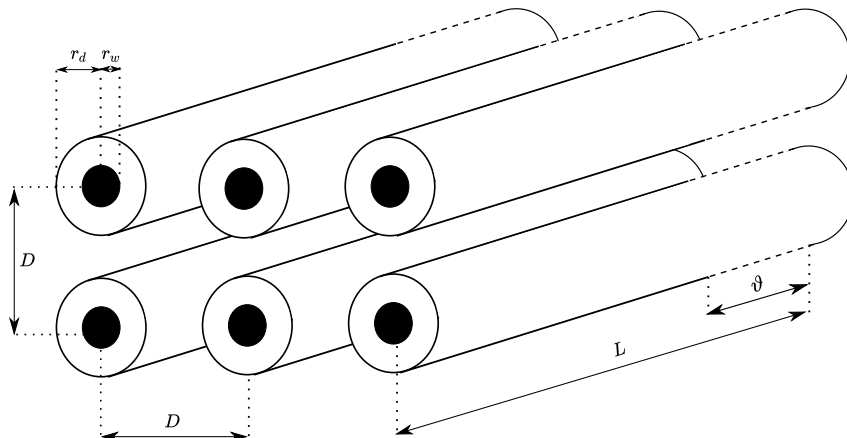


Fig. 4.10 Illustration of the partially coupled multi-conductor line used to test the proposed passivity verification and enforcement algorithm. Dimensions are not to scale.

#### 4.5.1 Partially coupled multi-conductor line

Our first example is a single-parameter electrical interconnect, depicted in Figure 4.10, composed of three adjacent differential pairs, each made of two parallel wires with a circular section (their center-to-center separation is 1.61 mm, while their length is 10 cm), forming a 6-wire multi-conductor transmission line. These differential pairs are assumed to be coupled for a reduced length  $L_c$ , which is our parameter  $\vartheta$ . The range of variation is  $\vartheta \in \Theta = [20, 40]$  mm. For additional details on this structure see [58]

Setting  $\bar{n} = 30$ ,  $\mathcal{I}_N = \mathcal{I}_D = 1, \dots, 5$ , we compute a first parameterized model using the PSK iteration over the set of training parameterized frequency responses; each is composed of  $\bar{k} = 500$  samples in the band [1 Hz, 5 GHz]. Since this initial model is not passive, we apply the proposed passivity enforcement algorithm that, in 12 iterations, produced a uniformly passive model. As illustrated in Figure 4.11 for a representative input/output response, the model is remarkably accurate with respect to the original data (the worst-case relative RMS error is  $13 \cdot 10^{-3}$ )

The original non-passive model is used to demonstrate the reliability issues of the method in [58] and, consequently, the improved performance of the proposed strategy. In the right panel of Figure 4.8 we show the results of the proposed adaptive sampling ( $J_{\max} = 10$ ), when applied to the model returned

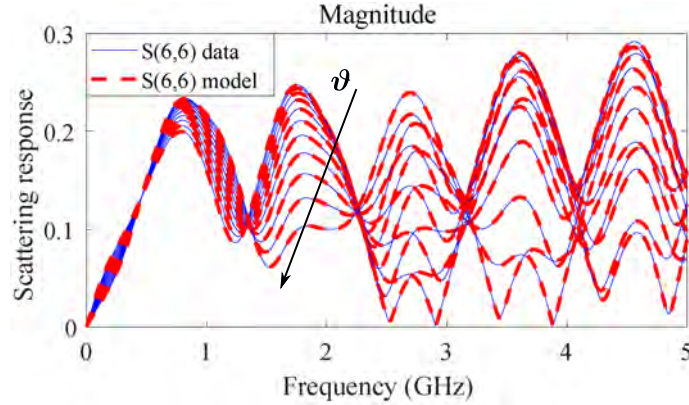


Fig. 4.11 Comparison between passive model and raw data for one representative response  $S_{6,6}(j\omega, \vartheta)$  of the partially coupled interconnect of Sec. 4.5.1. The various response pairs correspond to a linear sweep on  $\vartheta$  in its range. From [82] © 2018 IEEE

by the passivity enforcement method as implemented in [58]. The model still shows a localized violation, missed by the approach [58], whose results are depicted in the left panel.

## 4.5.2 A parameterized multi-board interconnect

In this second example we turn to a 2-parameters case. We consider a multi-board link connecting two multi-layer PCBs (see Figure 4.12 for an illustration and [3] for further details), by means of strip-line segments, a dedicated connector and its via fields. The geometrical dimensions of the link are parameterized by the radii of the via pad ( $\vartheta^1$ ) and anti-pad ( $\vartheta^2$ ) that, respectively, attain values in the ranges  $[100, 300] \mu\text{m}$  and  $[400, 600] \mu\text{m}$ .

The electrical behavior of this structure is known from  $\bar{m} = 81$  parametric frequency responses spanning the band  $[0, 5]$  GHz (courtesy of Prof. C. Schuster and Dr. J. Preibisch, Technische Universität Hamburg-Harburg, Hamburg, Germany). We constructed an initial model with  $\bar{n} = 24$ , and  $\mathcal{I}_N = \mathcal{I}_D$  defined by (2.62) by setting  $\bar{\ell}_1^N = \bar{\ell}_2^N = 3$ ,  $\bar{\ell}_1^D = \bar{\ell}_2^D = 3$ . The resulting worst-case relative RMS error among all the input/output responses is  $14 \cdot 10^{-3}$ . This model turns out to be not passive, as confirmed by top-left panel of Figure 4.13, where the red dots correspond to the non-passive parameter instances. We thus apply our proposed passivity enforcement algorithm.

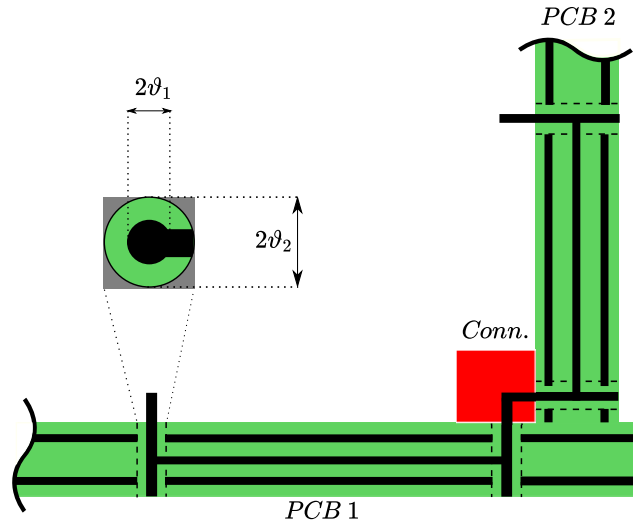


Fig. 4.12 High-speed interconnection between two multi-layer PCBs. The drawing is only for illustrative purposes and not to scale. From [86] © 2022 IEEE

The other panels of Fig. 4.13 show the intermediate iterative steps, confirming that the violation (red) areas are progressively removed. We obtained a passive model in 5 iterations (for a total CPU time of 12 minutes), whose worst-case relative RMS error is  $14 \cdot 10^{-3}$ ; this is remarkable, since no accuracy degradation occurred while enforcing passivity. The model accuracy is confirmed by the panels of Figure 4.14, comparing the responses of the passive model to the raw data, by sweeping one single parameter at a time, while keeping the other fixed.

### 4.5.3 A transmission-line network

The last example is a lossy transmission-line network represented in Figure 4.15, made of four cascaded line segments with three internal stubs, loaded with purely resistive loads. For this structure we consider two different and increasingly complex parameterizations. First, we parameterize the length of the central stub and of its adjacent line segments, that vary in the ranges  $\vartheta^1 \in [5, 7]$  mm and  $\vartheta^2 \in [9, 10]$  mm, respectively. Then, we consider a more complex 3-parameters version, obtained by parameterized also the load connected to the central stub by means of its reflection coefficient  $\vartheta^3 \in [0.1, 0.5]$ ; we additionally restricted the range of  $\vartheta^1$  to  $[6, 7]$  mm, since the first extended range led to a first-pass passive model, which required no further compensation. All the

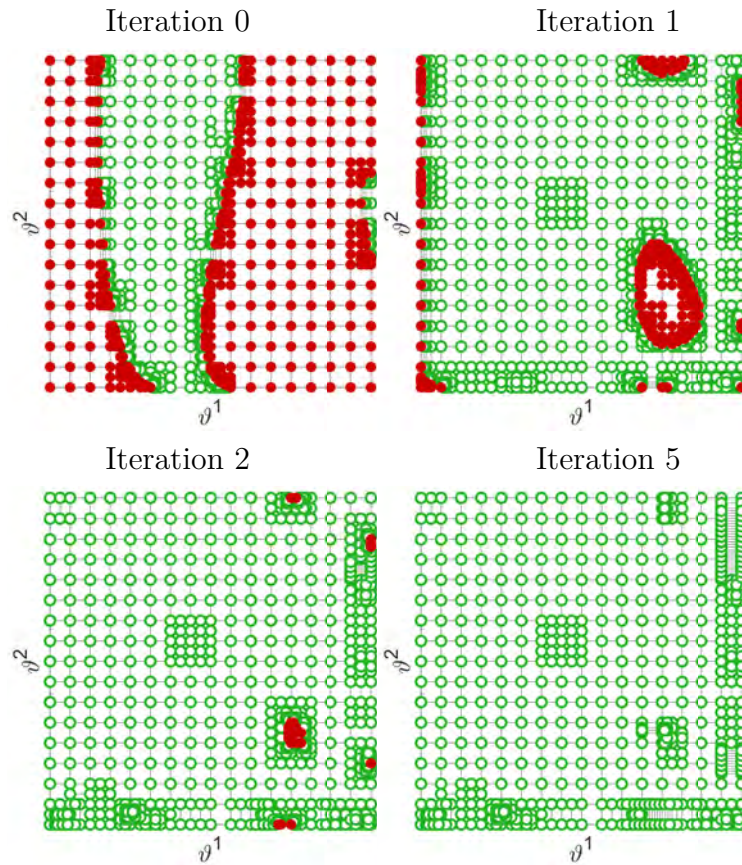


Fig. 4.13 Passivity characterization of the multiboard interconnect model illustrated after successive passivity enforcement (outer) iterations. Empty green (filled red) dots highlight passive (non-passive) parameter values for which no (some) SHH eigenvalues are detected by the proposed adaptive sampling algorithm. From [82] © 2018 IEEE

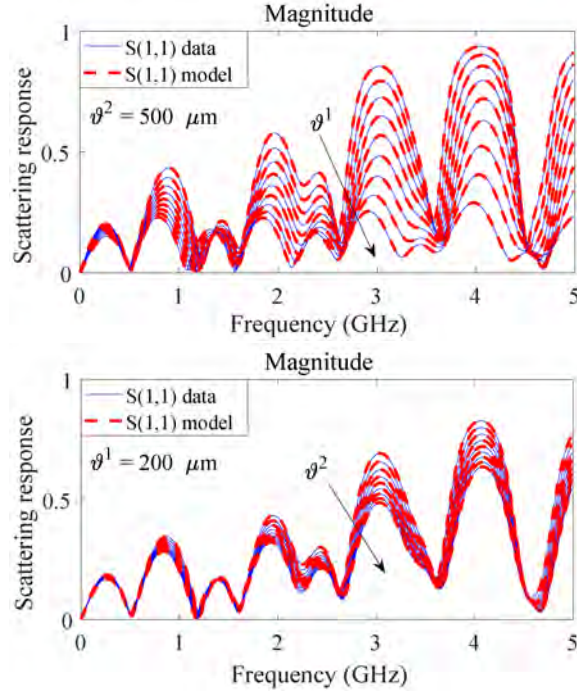


Fig. 4.14 Model responses compared to raw data for the multiboard interconnect parameterized by via pad ( $\vartheta^1$ ) and antipad ( $\vartheta^2$ ) radii. From [82] © 2018 IEEE

other geometric and electrical parameters are fixed to their nominal values:  $L = 7$  mm for the direct lines,  $L_s = 1$  mm for the stubs and  $\Gamma_1 = \Gamma_3 = 0.5$  for the other reflection coefficients.

Using the PSK scheme, we extracted two initial models: for the 2-parameters we set  $\bar{n} = 20$ ,  $\bar{\ell}_1^N = \bar{\ell}_2^N = 3$ ,  $\bar{\ell}_1^D = \bar{\ell}_2^D = 3$ , while for the 3-parameters we have  $\bar{n} = 20$ ,  $\bar{\ell}_1^N = \bar{\ell}_2^N = 3$ ,  $\bar{\ell}_3^N = 2$ , and  $\bar{\ell}_1^D = \bar{\ell}_2^D = 3$ ,  $\bar{\ell}_3^D = 2$ . The definitions of order sets  $\mathcal{I}_N$ ,  $\mathcal{I}_D$  is then obtained with (2.62).

The raw dataset is composed of parametric scattering responses, evaluated over the band [1 Hz, 10 GHz], that are obtained with a frequency-domain 2D integral field solver, in combination with a dedicated transmission line solver. With respect to the available validation data, the worst-case RMS relative error is  $46.2 \cdot 10^{-3}$  and  $22.6 \cdot 10^{-3}$  for bi- and tri-parameter models, respectively.

As confirmed by Figure 4.16, both the initial parameterized models were not passive. Applying the proposed passivity enforcement scheme, the 2-parameter model became passive in 9 iterations ( $J_{\max} = 8$ ), that required 22 minutes.

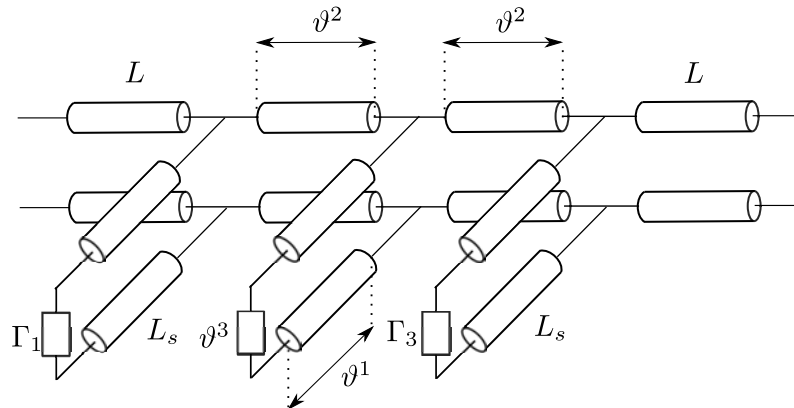


Fig. 4.15 Illustration of the transmission-line network used as test-case.

Instead, in the 3-parameter case, only 1 iteration (20 minutes) was needed to eliminate the small violation localized in the top right corner (see Figure 4.16).

The accuracy of the final passive 2-parameter model is confirmed in the panels of Figure 4.17a-b, that compare its responses with respect to the original raw data for fixed values  $\vartheta^1 = 6.1$  mm and  $\vartheta^2 = 9.5$  mm, while sweeping the other free parameters. Similarly, the panels c-b of Figure 4.17 confirm the accuracy of the passive 3-parameters model. The final worst-case RMS error is  $46.2 \cdot 10^{-3}$  and  $22.7 \cdot 10^{-3}$  for the 2- and 3-parameter models, respectively.

#### 4.5.4 Computational Times

Finally, we compare the computational performances (in terms of required CPU time) of the proposed algorithm with the method in [58]. We remark that this comparison is performed only for single-parameter models, for which the method[58] can be applied.

The results of this comparative test are shown in Figure 4.18, demonstrating that the CPU times of the proposed adaptive algorithm (orange bars) are comparable to (and in some cases even lower than) the ones of [58] (empty bars). Thus, the improvement in reliability comes with no extra cost in terms of computational effort.

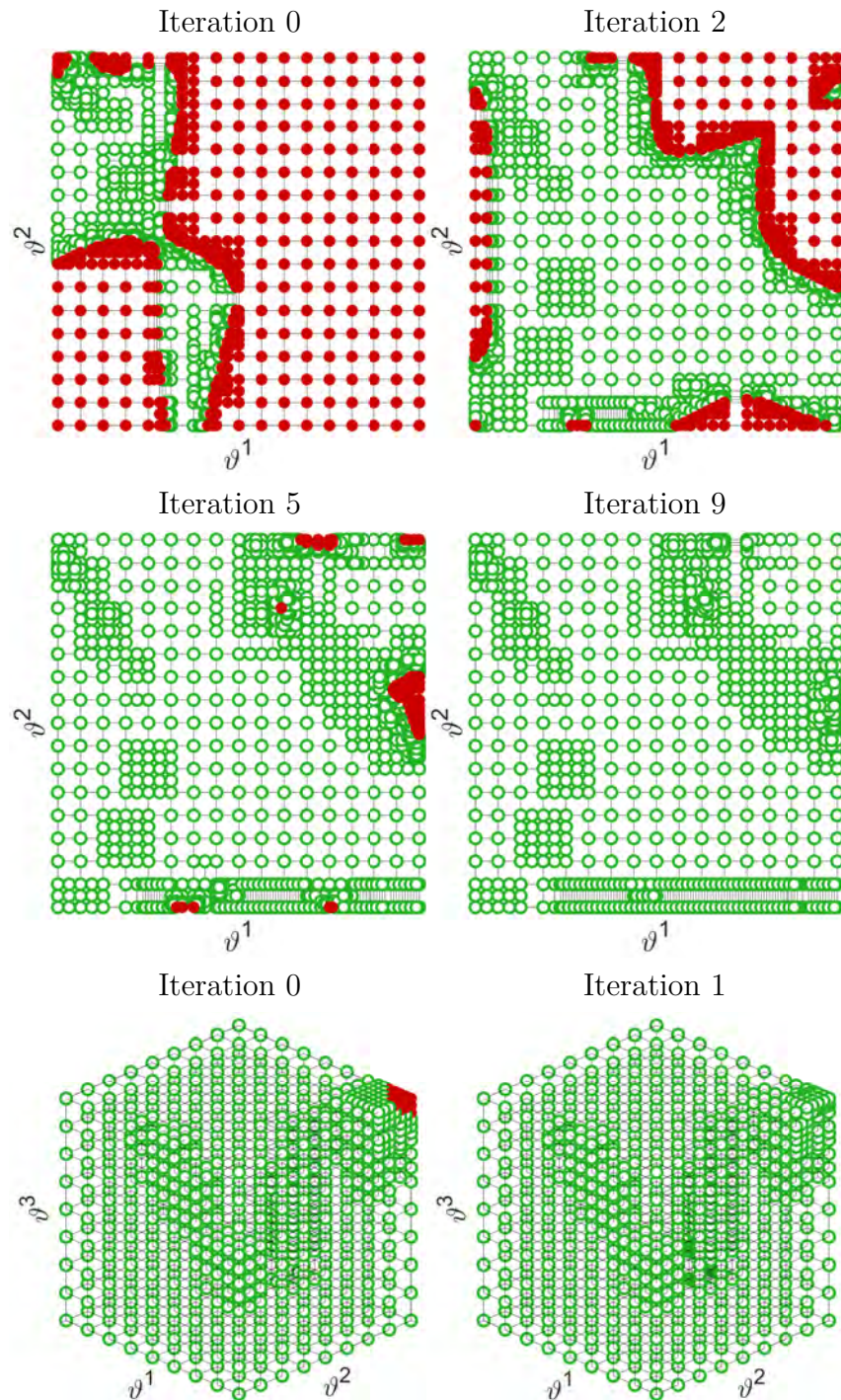


Fig. 4.16 As in Fig. 4.13, but for the transmission line network models. Panels (a)–(d) refers to the bi-variate model, while (e)–(f) show the parameter space for the tri-parameter case. From [82] © 2018 IEEE



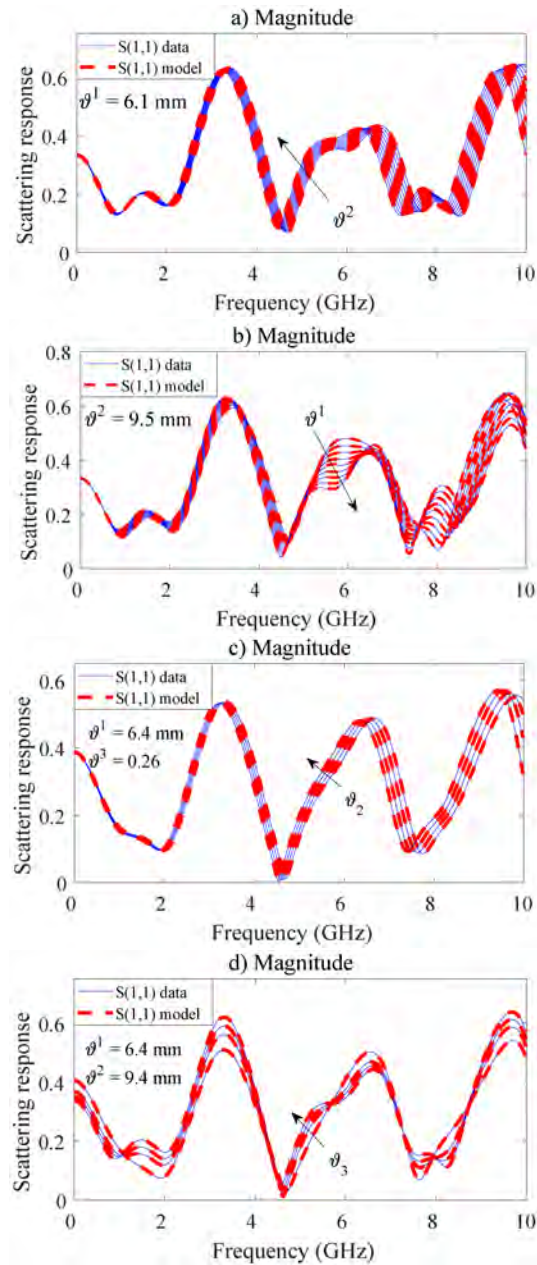


Fig. 4.17 a) and b): comparison between original data and passive model responses for the bi-parameter transmission line model, parameterized by stub and line lengths,  $\vartheta^1$  and  $\vartheta^2$  respectively; c) and d) same for the tri-parameter transmission line model, parameterized by stub and line lengths ( $\vartheta^1$ ,  $\vartheta^2$  respectively) and load reflection coefficient  $\vartheta^3$ . From [82] © 2018 IEEE.

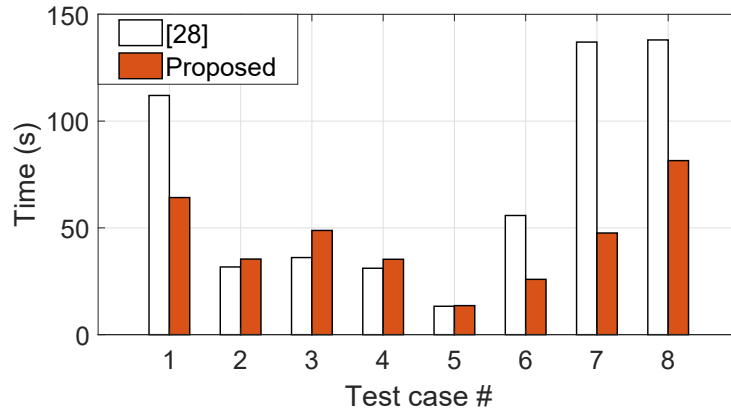


Fig. 4.18 Runtime of proposed passivity enforcement scheme (orange bars) compared to [58] (empty bars). From [82] © 2018 IEEE

## 4.6 Limitations and open problems

The proposed passivity verification and enforcement schemes has proven to be very effective in all the considered test-cases. Major reliability improvements and good scaling properties up to 3 independent parameters has been demonstrated with relevant numerical examples.

However, this approach can not be considered the ultimate solution to the problem of multi-variate passivity enforcement problem. In fact, although the computationally most expensive task is the computation of the Hamiltonian eigenvalues (whose complexity scales as  $O(n^3)$  [87], where the matrix dimension  $n$  scales linearly with respect to the number of ports and the dynamic/polynomial orders), the number of required eigenvalue extractions scales exponentially with the number of parameters, making the overall complexity to scale as  $O(k^\rho)$ ,  $k \in \mathbb{N}$ . On standard machines, the applicability of the proposed approach is therefore limited up to 2-3 parameters only.

The strategy devised in [86] partially overcomes this problem, exploiting an LMI-based approach that theoretically guarantees uniform passivity. The computational complexity of a LMI-constrained optimization problem (if no matrix structural properties are exploited) scales with the sixth power of the number of optimization variables [88], that are cumulatively determined by the number of parameters, dynamic/polynomial orders and the number of ports. In principle, the approach [86] has no limitations to the number of parameters,

as long as the number of poles, polynomial orders and number of ports are sufficiently small.

Figure 4.19 confirms the above theoretical discussion on a numerical test-case. We model a one-parameter multi-conductor transmission line (see Section 4.19 for details) for an increasing number of I/O ports, namely from 2 to 6 ports, while keeping fixed the number of poles and polynomial orders. It results that, for comparable model-data accuracies (Figure 4.19, top panel), the scaling law of the computational runtimes (Figure 4.19, bottom panel) of [86] is very unfavorable with respect to the one associated to the proposed adaptive sampling approach. In addition to the CPU effort, it should be noted that also the memory requirements of [86] do not scale favorably. As an example, the 6 ports model synthesis required almost 23 GB of RAM, while the implementation of the proposed algorithm can run on a standard 16 GB RAM laptop.

## 4.7 Adaptive stability enforcement

Let us now review the problem of enforcing uniform stability introduced in Section 4.1. Based on [4], we showed that the positive realness of the model denominator is a sufficient condition to guarantee uniform stability. We concluded Section 4.1 arguing that an appropriate selection of frequency-parameter locations where to enforce stability constraints is fundamental to practically employ this technique.

However, we now understand that requiring the denominator to be a positive real function is equivalent to asking for its passivity, when interpreted as an immittance one-port system. The presented adaptive passivity verification algorithm can thus be directly applied to efficiently search for the frequency-parameter locations where the denominator is not passive, thus not PR. Assuming to be at the  $\nu$ -th PSK iteration, we can therefore set up an adaptive stability enforcement algorithm as:

1. we extract the denominator  $D^{\nu-1}(s, \boldsymbol{\vartheta})$  associated to the model  $\mathbf{H}^{\nu-1}(s, \boldsymbol{\vartheta})$  evaluated at the previous iteration  $\nu - 1$ ;

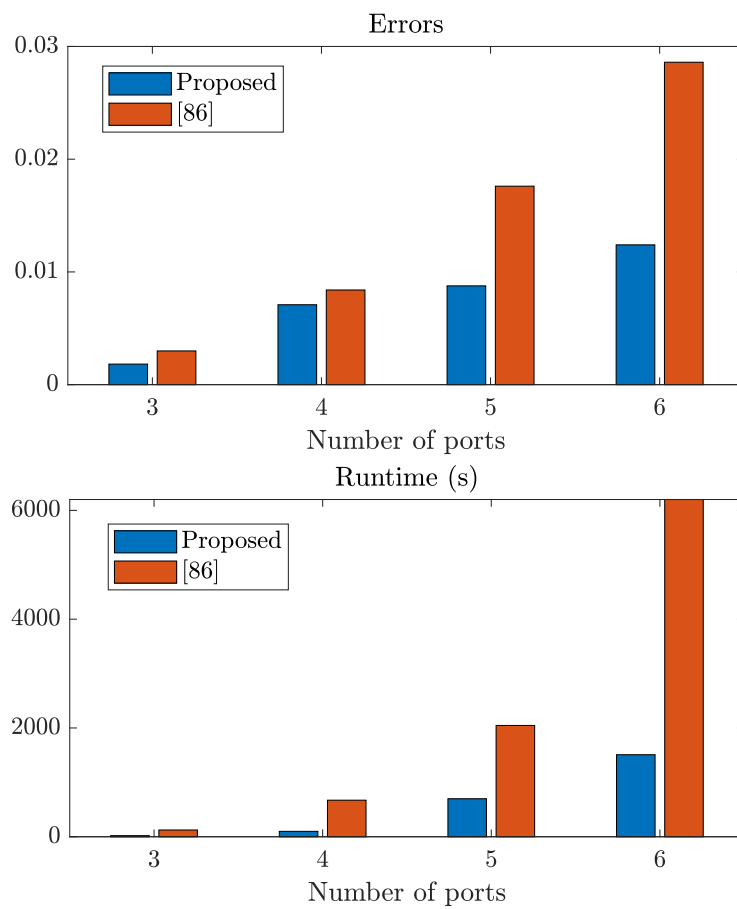


Fig. 4.19 Comparison of the proposed technique (blue bars) with respect to [86] (orange bars). Top panel: modeling errors; bottom panel: CPU runtimes.

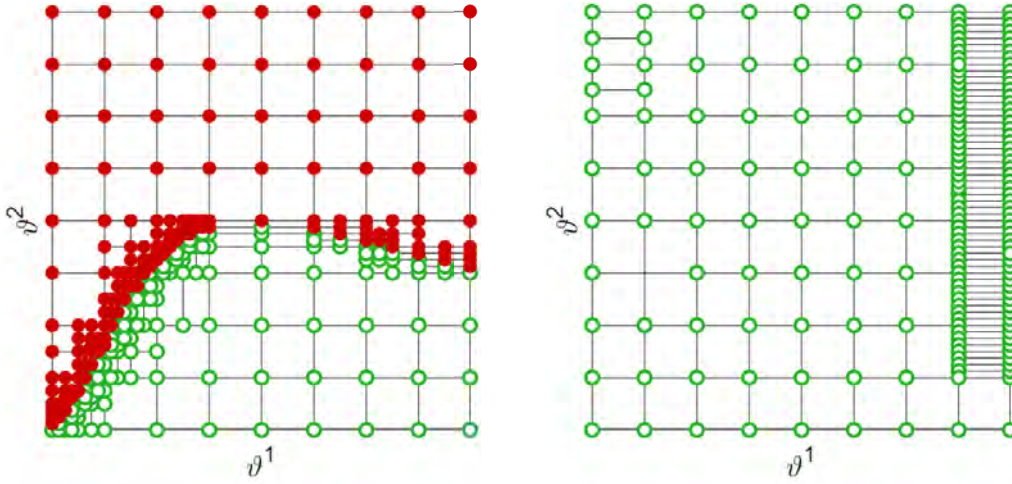


Fig. 4.20 Adaptive sampling of a 2D parameter space at two successive PSK iterations. Red and green dots denote, respectively, locally unstable and stable parameter combinations. Left panel: residual local unstable parameter combinations are present during iteration. Right panel: uniformly stable macromodel obtained by adding uniform stability constraints at locally unstable parameter points.

2. assuming  $D^{\nu-1}(s, \boldsymbol{\vartheta})$  to be an immittance one-port system, we perform the adaptive passivity check to gather the locations  $(j\omega_q, \boldsymbol{\vartheta}_q) \in \mathcal{A}^\nu$  where  $D^{\nu-1}(s, \boldsymbol{\vartheta})$  is not passive, i.e., not PR;
3. based on the set  $\mathcal{A}^\nu$ , we build the PR-ness constraints (4.4) and obtain the current denominator  $D^\nu(s, \boldsymbol{\vartheta})$  solving the constrained PSK problem (4.5).

This procedure has been first introduced in [4] in case of a single independent parameter; in [89] the presented adaptive passivity verification algorithm has been employed to extend the applicability of this method to higher dimensional spaces.

Figure 4.20 reports the results of the proposed adaptive sampling method applied to enforcing uniform stability to the 2-parameters transmission line network model of Section 4.5.3: the left panel shows localized residual (possibly) unstable regions (red dots) while, on the right panel, we show the results obtained by enforcing the uniform stability constraints in correspondence to locally not PR parameter locations. The absence of residual not PR regions demonstrate the capability of this approach to synthesize uniformly stable models.

Although the use of the adaptive sampling scheme dramatically reduces the computational effort required to formulate the stability constraints (with respect to a fine gridding), we must be aware of its limitations (see Section 4.6), that make it not adequate in case of medium to large parameter spaces. This problem will be addressed in Chapter 5, Section 5.2 where, under some conditions on the parameter basis functions, we will introduce a stability enforcement strategy whose complexity is independent on the number of parameters.

### Robust stability enforcement

The above procedure has proved its effectiveness in many documented test-cases [4, 90]. However, we should keep in mind that uniform stability is enforced only in correspondence to the selected frequency parameter locations  $(j\omega_q, \boldsymbol{\vartheta}_q)$ , and nothing prevents the current denominator estimate  $D^\nu(s, \boldsymbol{\vartheta})$  to become negative at different parameter locations at the subsequent iteration. An example of this occurrence is reported in the left column of Figure 4.21, that illustrates the stability violation locations on the parameter space for three successive fitting iterations for the case study of a parameterized H-antenna we originally presented in [91] (full details can be found in [44]). A red to black relative color scale is used to represent the extent of each local violation (in terms of minimum real part of the model denominator). It can be noted that, despite the constraints (4.4) are enforced at each of the highlighted points, new violations reappear at the same locations at subsequent iterations.

Such a circumstance may dramatically impair the PSK scheme convergence, in which case we expect that the coefficient update  $d_{n,\ell}^\nu$  vanishes through subsequent iterations, as

$$d_{n,\ell}^\nu = |r_{n,\ell}^\nu - r_{n,\ell}^{\nu-1}| \xrightarrow{\nu \rightarrow \infty} 0 \quad \forall n, \forall \ell. \quad (4.39)$$

The unpredictable re-appearance of previously compensated violation may cause the denominator estimates  $r_{n,\ell}^\nu$  to undergo erratic behaviors through the PSK iterations, which never converge as required by (4.39). In this regard, we report in Figure 4.22 the results associated with the same H-antenna example of Figure 4.21; despite the model responses, (mid-left panel of Fig. 4.22) are accurate, the top-left panel highlights that the denominator estimates never

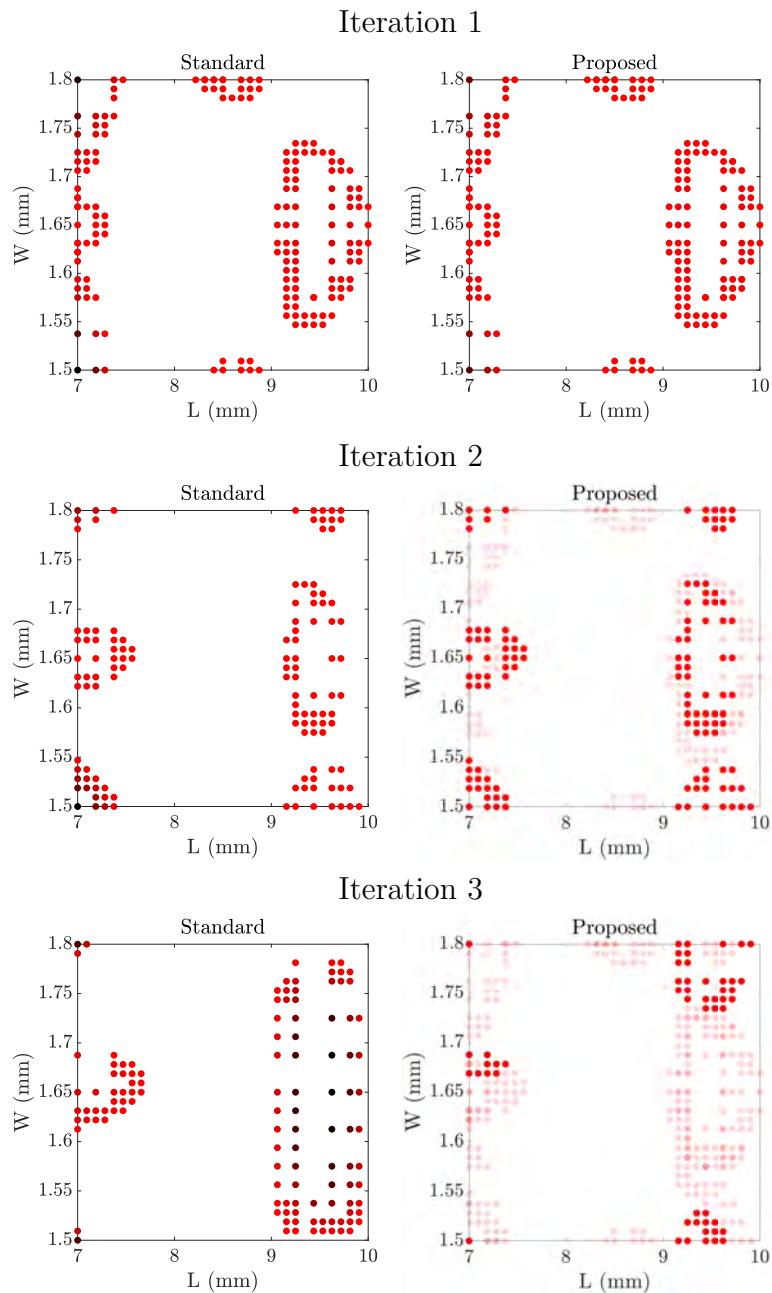


Fig. 4.21 Location in the parameter space  $\Theta$  of stability constraints for three successive fitting iterations. Left column: standard approach of [90] with independent stability constraints at each iteration. Right column: proposed robust approach, with constraints accumulation through iterations (violation extent is represented with a red-to-black color scale, with transparent dots representing the constraints at previous iterations). From [91] © 2019 IEEE

stabilize leading, in turn, to positive instances of the parameterized model poles (bottom-left panel).

We tackled this problem in [91] where, inspired by the *robust* passivity enforcement implementation [92], we introduced a robust version of the stability preserving PSK iteration. The direct implementation of the approach [92], would require to aggregate *predicted* constraints in order to prevent the re-appearance of PR-ness violations. However, such a *look ahead* scheme necessitates of (possibly very expensive) additional PSK iterations, only devoted to predict the future constraints' locations.

In [91] we proposed, instead, a more efficient *look behind* robust iteration. Assuming to be at the  $\nu$ -th PSK iteration, the stability constraints are built over the aggregated set of previous violation locations, as

$$\Re\{D^\nu(j\omega, \boldsymbol{\vartheta})\} > \gamma, \quad \forall(j\omega_q, \boldsymbol{\vartheta}_q) \in \cup_{i=1}^{\nu-1} \mathcal{A}^i, \quad (4.40)$$

The results of the proposed robust iteration are illustrated in the right column of Figure 4.21, where each panel shows the accumulated constraints from the previous iterations using transparent dots. This example provides a clear illustration that, from a geometrical standpoint, aggregating the constraints is equivalent to narrowing the feasible set of the optimization problem, that is constructed as the intersection of the feasible sets corresponding to the previous iterations.

In Figure 4.22 we show that, using the aggregate set of constraints, the denominator coefficient estimates converge monotonically (top panel), the model accuracy is not compromised with respect to the standard approach (middle panel), and the parameterized poles are uniformly confined in the left-half complex plane (bottom panel).



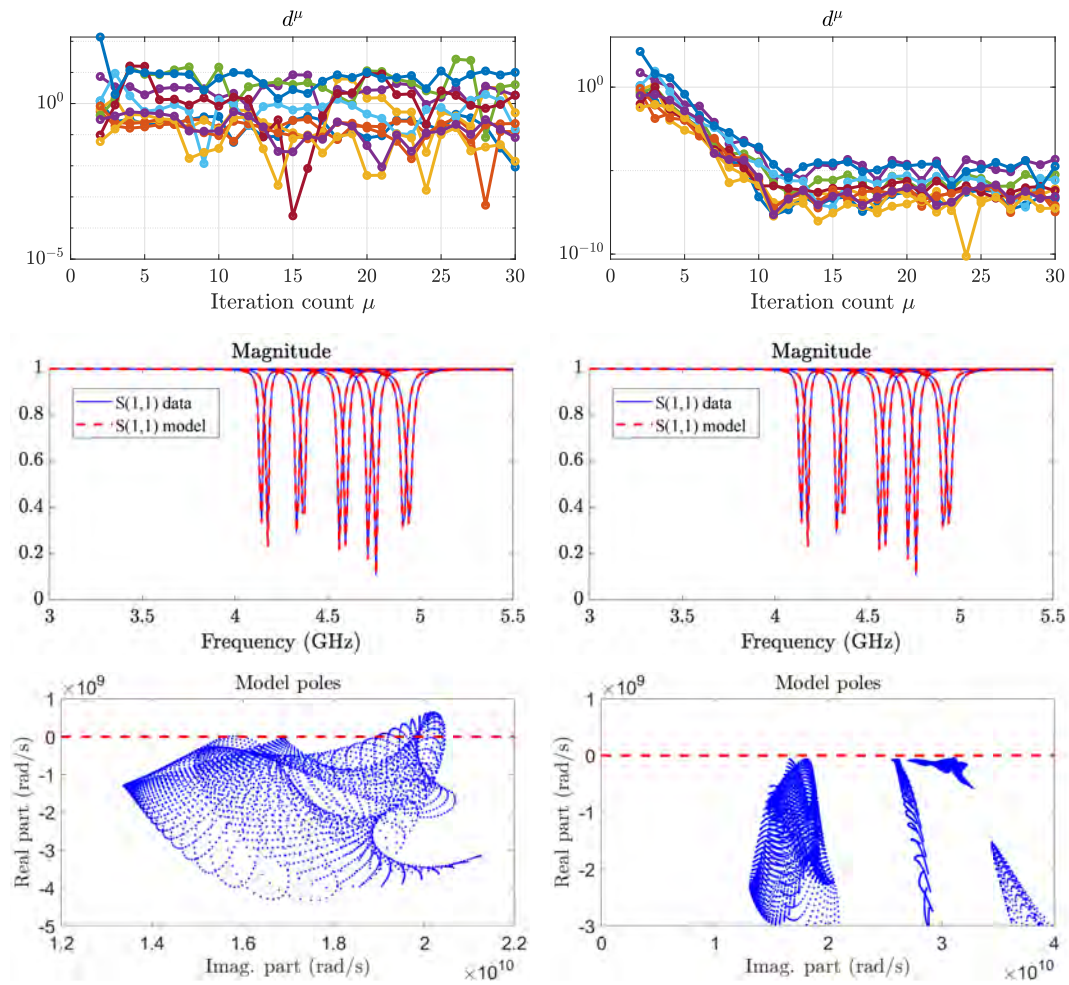


Fig. 4.22 Macromodeling results using standard PSK iteration (left panels) and proposed look-behind robust iterations (right panels) applied to an H-shaped antenna. Top panels: evolution of coefficient updates through iterations. Middle panels: selected model responses for few parameter configurations, compared with raw data. Bottom panels: parameterized model poles. From [91] © 2019 IEEE

# Chapter 5

## Towards High-dimensional parameterized macromodeling

In the last few decades, high-dimensionality problems have been unceasingly gaining more importance, mainly for their relevance in practical applications. High-dimensionality problems are typical of many scientific fields, ranging from applied mathematics and different areas of science (modeling of scattered data and solutions of partial differential equations) to, for instance, mathematical finance, computer graphics and learning theory [93].

In our context, the availability of high-dimensional macromodels that depend on several independent parameters would be highly beneficial, especially for the designers, who can exploit these models for simplified and more efficient design centering, optimization and sensitivity analyses.

It has been already noted that the complexity of the PSK parameterized macromodel as defined in Chapter 2, Section 2.3.3, does not scale favorably in case of a large number of parameters. Taking inspiration from the vast literature concerning the multi-variate approximation theory, in this chapter we are going to introduce a novel parameterized macromodel structure that allows a much better scaling in higher dimensional settings.

## 5.1 Radial Basis Functions approximation

### 5.1.1 Scattered data approximation

Let us take a step back to review the problem of scattered data interpolation. Assume that we are given with *scattered* instances  $\check{s}(\mathbf{x}_i)$ ,  $i = 1, \dots, N$  of an unknown (smooth) function  $\check{s}(\mathbf{x}) : \Theta \subseteq \mathbb{R}^\rho \rightarrow \mathbb{R}$ ,  $\rho \gg 1$ , that we want to reconstruct up to some prescribed accuracy. The solution to such an *approximation* problem belongs to the class of *mesh-free* or *unstructured* methods. Briefly speaking, a mesh-free approximation technique does not require the data to be arranged according to a predefined “rigid” structure (e.g., Cartesian lattice). These techniques have been designed to guarantee an improved versatility that enables to easily solve high-dimensional problems, for which no a-priori data distributions are given.

In this regard, the main feature that makes mesh-free approximations so interesting, is that their complexity does not depend on the space dimension  $\rho$ , but only on the smoothness of the function  $\check{s}(\mathbf{x})$  [93–96]. More in details, given the target function instances  $\check{s}(\mathbf{x}_i)$  evaluated in correspondence of data sites  $\mathbf{x}_i$ , we set up the approximation  $s(\mathbf{x}) \approx \check{s}(\mathbf{x})$ , as

$$s(\mathbf{x}) = \sum_{\ell=1}^{\bar{\ell}} \alpha_\ell \Phi_\ell(\mathbf{x}) \quad (5.1)$$

where  $\alpha_\ell$  are expansion coefficients (to be estimated) and  $\Phi_\ell : \mathbb{R}^\rho \rightarrow \mathbb{R}$  are some basis functions in charge of mapping the  $\rho$ -dimensional domain to the scalar image of the approximation. The unknown expansion coefficients  $\alpha_\ell$  are computed enforcing the accuracy of the model  $s(\mathbf{x})$  on the data samples  $\check{s}(\mathbf{x})$ ; this translates in the solution of a linear system  $\mathbf{\Phi}\mathbf{a} = \check{\mathbf{s}}$  (or least squares problem  $\mathbf{\Phi}\mathbf{a} \approx \check{\mathbf{s}}$ ) where  $[\mathbf{\Phi}]_{i,\ell} = \Phi_\ell(\mathbf{x}_i)$ ,  $a_\ell = \alpha_\ell$  and  $\check{s}_i = \check{s}(\mathbf{x}_i)$ .

### 5.1.2 Radial Basis Functions

In order to obtain the desired results, both in terms of accuracy and scalability, an appropriate choice of the basis functions  $\Phi$  is fundamental. In literature, it is common to set  $\Phi_\ell(\mathbf{x})$  as *Radial Basis Functions* (RBFs), initially introduced by

R.L. Hardy in [97], which have proven to be extremely powerful. Following the definition provided in [93], they are defined as the composition of a univariate scalar function  $\phi(\|\mathbf{x} - \mathbf{x}_\ell\|)$  that depends on the norm (usually Euclidean) of the input variable  $\mathbf{x}$ , as

$$\Phi_\ell(\mathbf{x}) = \Phi(\mathbf{x} - \mathbf{x}_\ell) = \phi(\|\mathbf{x} - \mathbf{x}_\ell\|) \quad (5.2)$$

The input variable  $\mathbf{x}$  thus appears inside a suitable *norm* operator, that translates the multi-variate input into a scalar variable. This is the reason behind the excellent scalability properties of the RBFs: independently of the dimension  $\rho$  of the input space  $\Theta$ , the radial basis function always “processes” nothing more than a scalar variable.

Following definition (5.2), the RBFs are defined upon the *hyper-parameter*  $\mathbf{x}_\ell$ . Although it may have several geometrical interpretations, we will refer to  $\mathbf{x}_\ell$  as the *center* of the radial basis function. These basis functions are termed *radial* in the sense that  $\phi(\|\mathbf{x} - \mathbf{x}_\ell\|) = \phi(r)$ , where  $r$  can be thought as the “radius” of the basis function, as it attains the same value at a distance  $r$  from its symmetry center  $\mathbf{x}_\ell$ . See Figure 5.1 for a graphical illustration. In these terms, the functional form of the approximation (5.1) can be interpreted as a weighted sum of *shifted* and *scaled* radial basis functions.

**Remark 13.** *The Radial Basis Functions we will use throughout this chapter have strong connections with other mathematical fields. For instance, in functional analysis they serve as kernels of integral operators. In the machine learning community (especially in case of Support Vector Machines) it is common to use RBF as “kernels” to map the input space to the feature space. For these reasons, in literature it is common to interchangeably use the terms kernel and Radial Basis Functions.*

### 5.1.3 Mairhuber-Curtis Theorem

Although slightly outside the scope of this work, we deem useful to provide additional details about mesh-free methods in scattered data interpolation. In particular, studying the interpolation problem  $\Phi \mathbf{a} = \mathfrak{s}$ , it is possible to prove that we can guarantee the existence and uniqueness of the solution  $\mathbf{a}$  only if

we employ unstructured methods [93, 95]. The Reader who is not interested can skip this part without loss of continuity.

In details, it is known that a solution  $\mathbf{a}$  exists and is unique if and only if  $\Phi$  is invertible, i.e.,  $\det\{\Phi\} \neq 0$ . This condition always holds in the univariate setting, in which one can fix in advance a basis function (such as a degree  $N - 1$  polynomial) to exactly interpolate  $N$  distinct (real-valued) data points. In the multivariate setting, however, the *Mairhuber-Curtis* theorem shows that we can not draw similar conclusions. Before presenting the theorem, let us introduce the following definition

**Definition 6** (*Haar space*). *Let us assume that  $\Theta \subseteq \mathbb{R}^\rho$  contains at least  $\bar{\ell}$  points. In addition, let us define  $B \subseteq C(\Theta)$ , with  $C(\Theta)$  the space continuous functions on  $\Theta$ , as a  $\bar{\ell}$ -dimensional linear space with basis  $\{\Phi_1(\mathbf{x}), \dots, \Phi_{\bar{\ell}}(\mathbf{x})\}$ . Then,  $B$  is a  $\bar{\ell}$ -dimensional 'Haar' space on  $\Theta$  if*

$$\det(\Phi) \neq 0 \tag{5.3}$$

for any distinct  $\mathbf{x}_1, \dots, \mathbf{x}_{\bar{\ell}} \in \Theta$ .

The existence of a Haar space on  $\Theta$  guarantees the invertibility of the system matrix  $\Phi$ . We have the following result (which we report *verbatim* from [95])

**Theorem 4** (*Mairhuber-Curtis*). *Suppose that  $\Theta \subseteq \mathbb{R}^\rho$ ,  $\rho \geq 2$ , contains an interior point. Then there exists no Haar space on  $\Theta$  of dimension  $\bar{\ell} \geq 2$*

Theorem 4 states that, if we define in advance a set of basis functions  $\{\Phi_1(\mathbf{x}), \dots, \Phi_{\bar{\ell}}(\mathbf{x})\}$  that build  $\Phi$ , we can not guarantee its invertibility, thus the existence of a multi-dimensional interpolant of the form 5.1. As an example, multi-variate polynomials can not construct Haar spaces, thus can not be used to interpolate data that are arbitrarily scattered on  $\Theta$  (see [95] for details).

Therefore, the Mairhuber-Curtis Theorem states that a multi-dimensional interpolation problem should not rely on fixed basis functions  $\Phi_\ell(\mathbf{x})$ ; instead, they must be data-dependent, so that an invertible system matrix  $\Phi$  can be constructed, independently on the data locations. In this view, the mesh-free methods can be considered the natural framework for multi-variate interpolation problems.

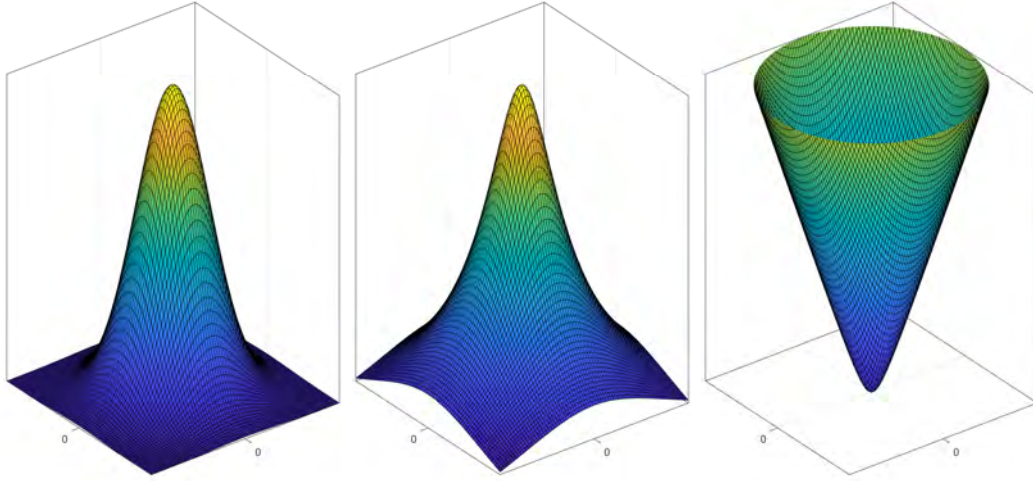


Fig. 5.1 Three radial basis functions tested in this work, depicted in a 2D domain. From left to right: Gaussian RBF, inverse multiquadric RBF, multiquadric RBF. From [101] © 2020 IEEE

#### 5.1.4 RBF taxonomy

In the literature, it is common to subdivide the radial basis functions in two main categories: infinitely smooth and compactly supported RBFs. The latter have been originally proposed in [98–100] with the purpose of constructing *sparse* RBF interpolation matrices. Since we do not require any sparsity in our problems, we will employ infinitely smooth RBFs. The most notable are:

- Gaussian:  $\Phi_\ell^\varepsilon(\mathbf{x}) := e^{-\varepsilon^2 \|\mathbf{x} - \mathbf{x}_\ell\|^2}$
- Multiquadric:  $\Phi_\ell^\varepsilon(\mathbf{x}) = \sqrt{\varepsilon^2 + \|\mathbf{x} - \mathbf{x}_\ell\|^2}$
- Inverse multiquadric:  $\Phi_\ell^\varepsilon(\mathbf{x}) = \frac{1}{\sqrt{\varepsilon^2 + \|\mathbf{x} - \mathbf{x}_\ell\|^2}}$

Other notable RBFs are, for instance: inverse quadric, thin plate spline and Matern kernels. See [94] for more details.

In Section 5.1.4 we will analyze the former three, when used to embed the parameter variability in frequency-domain parameterized macromodels; we provide a graphical representation of these RBFs in Figure 5.1. All three of

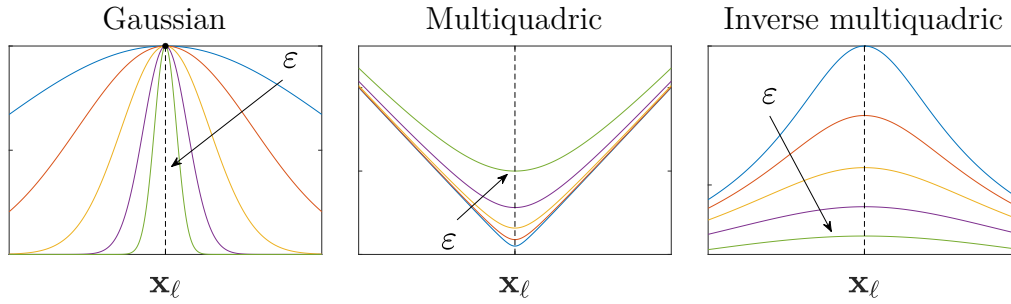


Fig. 5.2 In the three panels, we illustrate the effect that variations of the shape parameter have on the three considered RBFs. For ease of representation, we represent the RBFs evaluated along a one one-dimensional cut of the parameter space.

these functions are positive over their domain; in Section 5.2, we will exploit this property to show that, when used as parameter basis functions, it is possible to guarantee the model uniform stability through a finite set of constraints formulated on the model coefficients.

We highlight that most of the RBFs are defined upon a hyper-parameter  $\varepsilon$ , commonly known as *shape parameter* or *form factor*, that determines their “shape”; in particular, this parameter selects the “width” or “size” of the RBFs, that become narrower or flatter depending on its value. Figure 5.2 shows how variations of the RBFs’ form factor affect their shape.

In order to obtain the desired approximation accuracy, this parameter must be optimized depending on the problem at hand; unfortunately, the problem of choosing the optimal value is strongly non-convex and, to the best of the Author’s knowledge, all the available methods are based on heuristic rules and do not provide guarantees of optimality.

We can thus conclude that most common radial basis functions are fully defined up to the selection of two hyper-parameters: the RBFs’ centers  $\mathbf{x}_\ell$  and the shape parameter  $\varepsilon$ . The selection of sub-optimal hyper-parameters will be subject of the investigations of Chapter 6.

## 5.2 High-dimensional macromodeling

Planning to use a mesh-free (unstructured) approach to embed several parameters in our parameterized macromodels is in fact quite straightforward.

Recalling the polynomial-based PSK model structure (2.63), we embed the parameter variability in the model by setting the RBFs as parameter basis functions, i.e.,

$$\xi_\ell^\varepsilon(\boldsymbol{\vartheta}) = \Phi_\ell^\varepsilon(\boldsymbol{\vartheta}) \quad (5.4)$$

which leads to the following *high-dimensional* PSK model structure

$$\mathbf{H}(s; \boldsymbol{\vartheta}) = \frac{\mathbf{N}(s; \boldsymbol{\vartheta})}{D(s; \boldsymbol{\vartheta})} = \frac{\sum_{n=0}^{\bar{n}} \sum_{\ell=1}^{\bar{\ell}_N} \mathbf{R}_{n,\ell} \xi_\ell^\varepsilon(\boldsymbol{\vartheta}) \varphi_n(s)}{\sum_{n=0}^{\bar{n}} \sum_{\ell=1}^{\bar{\ell}_D} r_{n,\ell} \xi_\ell^\varepsilon(\boldsymbol{\vartheta}) \varphi_n(s)}. \quad (5.5)$$

It should be noted that, although Equations (2.63) and (5.5) seem identical, the latter is independent of the dimension  $\rho$  of the embedding space  $\Theta$ . Indeed, in the standard model form (2.63) the strong dependence on the number of parameters is hidden in the definitions of sets  $\mathcal{I}_N$  and  $\mathcal{I}_D$ , whose cardinality grows exponentially as  $\rho$  increases. Conversely, in Equation (5.5), the number of basis functions  $\bar{\ell}_N, \bar{\ell}_D$  depend solely on the smoothness of the parametric variability and not on number of parameters. In the following, we will often refer to  $\bar{\ell}_N$  and  $\bar{\ell}_D$  as *parametric order*.

### High-dimensional model identification

Although the high-dimensional model form (5.5) slightly differs from the standard (2.63), the PSK iteration can be still adopted to identify the unknown coefficients. We therefore iteratively solve the linearized least-square problem  $\|\Psi^\nu \mathbf{c}^\nu\|_2 \approx 0$  over the  $m^t$  elements of the training set  $\mathcal{T}$ , as in Section 2.3.3.

In this regard, we deem useful to briefly recall that the iteration dependent regressor  $\Psi^\nu$  is defined as

$$\Psi^\nu = \begin{pmatrix} \Gamma^\nu & & \Xi_{(1,1)}^\nu \\ & \ddots & \vdots \\ & & \Gamma^\nu & \Xi_{(P,P)}^\nu \end{pmatrix} \quad (5.6)$$

For further use, we redefine  $\Psi^\nu$  as a function of the shape parameter, as  $\Psi^\nu(\varepsilon)$ . The dependence upon  $\varepsilon$  is inherited by its building blocks  $\Gamma^\nu = \Gamma^\nu(\varepsilon)$  and



$\Xi_{i,j}^\nu = \Xi_{i,j}^\nu(\varepsilon)$  that, following (2.69), are defined as

$$\mathbf{\Gamma}^\nu(\varepsilon) = \mathbf{W}^{\nu-1} \mathbf{X}_N(\varepsilon), \quad (5.7a)$$

$$\Xi_{(i,j)}^\nu(\varepsilon) = -\mathbf{W}^{\nu-1} \check{\mathbf{H}}_{(i,j)} \mathbf{X}_D(\varepsilon). \quad (5.7b)$$

In turn, matrices  $\mathbf{X}_{N,D}(\varepsilon)$  inherit the dependence on the shape parameter by the *kernel* matrices<sup>1</sup>  $\mathbf{K}_{N,D}(\varepsilon)$ , as

$$\mathbf{X}_{N,D}(\varepsilon) = \mathbf{\Phi} \otimes \mathbf{K}_{N,D}(\varepsilon) \quad (5.8)$$

where  $\mathbf{\Phi}$  collects the evaluated partial fraction basis as in (2.28) and  $\mathbf{K}_{N,D}(\varepsilon)$  is defined as

$$\mathbf{K}_{N,D}(\varepsilon) = \begin{pmatrix} \xi_1^\varepsilon(\boldsymbol{\vartheta}_1) & \cdots & \xi_\ell^\varepsilon(\boldsymbol{\vartheta}_1) \\ \vdots & & \vdots \\ \xi_1^\varepsilon(\boldsymbol{\vartheta}_{m^t}) & \cdots & \xi_\ell^\varepsilon(\boldsymbol{\vartheta}_{m^t}) \end{pmatrix} \quad (5.9)$$

We assume that the RBFs  $\xi_\ell^\varepsilon(\boldsymbol{\vartheta})$  used to construct matrices  $\mathbf{K}_N(\varepsilon)$  and  $\mathbf{K}_D(\varepsilon)$  are centered in correspondence of parameter values  $\boldsymbol{\vartheta}_\ell$ , collected in set  $\mathcal{C}_N$  and  $\mathcal{C}_D$ , respectively. In addition, we also assume that the RBFs are always positioned in correspondence to a subset of the available training samples, thus  $\mathcal{C}_N, \mathcal{C}_D \subseteq \mathcal{T}$ .

We conclude highlighting that the proposed high-dimensional version of the PSK can be naturally cast in the more efficient *Fast-PSK*, introduced in Section 2.3.3. For more details see [103].

### 5.2.1 Enforcing uniform stability

In Chapter 4, Sec. 4.7, we discussed the problem of enforcing uniform stability on parameterized models, i.e., guaranteeing that the parameterized model poles have a uniformly negative real part,  $\forall \boldsymbol{\vartheta} \in \Theta$ .

We already pointed out that directly operating on the model poles in the optimization step is not possible, and that it is preferable to pursue an indirect approach. In details, reporting the results of [4], we demonstrated that a

<sup>1</sup>In the context of kernel-based approximation, this matrix is also known as *Gram* matrix [102].

sufficient condition for uniform stability is the *positive realness* of the model denominator, which can be easily enforced in the fitting step.

However, to guarantee the positive realness of the denominator, we need to formulate local positivity constraints in correspondence to the frequency-parameter combinations where the denominator response is *negative real*, thus discretizing a continuous (possibly high-dimensional) space. This procedure inevitably suffers from the curse of dimensionality and can not be practically employed in our high-dimensional case.

In this regard, one of the main results of this work is indeed the formulation of uniform stability constraints that are independent of the number of parameters. The key enabling factor to overcome the curse of dimensionality is to formulate stability constraints that are based on the *model structure*, rather than upon the denominator's responses. As we preliminarily introduced in [104], then formally shown in [101], this is possible thanks to the use of positive semi-definite parameter basis functions

$$\xi_\ell^\varepsilon(\boldsymbol{\vartheta}) \geq 0, \quad \forall \boldsymbol{\vartheta} \quad (5.10)$$

as the three RBFs that we have taken in consideration. Under this assumption, it is possible to formulate a set of stability constraints directly on the denominator coefficients. Assuming  $r_{n,\ell}$  and  $r'_{n,\ell} + jr''_{n,\ell}$  to be the coefficients associated, respectively, to the real  $q_n$  and complex conjugate  $\alpha_n \pm j\beta_n$  stable poles, we have the following result, whose proof is available in Appendix A

**Theorem 5.** *Let  $\mathbf{H}(s, \boldsymbol{\vartheta})$  be defined as in (5.5). Assuming that the parameter basis functions are positive semi-definite and that the basis poles  $q_n$  of the partial fractions  $\varphi_n(s)$  are asymptotically stable, if the following conditions hold*

$$\begin{cases} r_{n,\ell} > 0 \\ -\alpha_n \cdot r'_{n,\ell} \pm \beta_n \cdot r''_{n,\ell} > 0 \end{cases} \quad (5.11)$$

*the model  $\mathbf{H}(s, \boldsymbol{\vartheta})$  is uniformly stable over  $\mathbb{R}^p$ .*

In case strict positivity is required, we can impose the more stringent condition

$$\begin{cases} r_{n,\ell} > \gamma_r \\ -\alpha_n \cdot r'_{n,\ell} \pm \beta_n \cdot r''_{n,\ell} > \gamma_c \end{cases}, \quad \text{with } \gamma_r, \gamma_c > 0 \quad (5.12)$$

which is numerically preferable.

The use of these constraints is better suited in the high-dimensional setting, since:

- it does not require to sample the frequency-parameters space, and
- the total number of constraints is defined a-priori by the selected model structure, and it is equal to  $\bar{\ell}_D(1 + \bar{n}_r + 2\bar{n}_c)$ .

We remark that the use of the above stability constraints is not restricted to the case of RBFs, as any positive semi-definite (structured or unstructured) parameter basis function can benefit from these new constraints.

As an example, we report the case study we presented in [101] built around the use of *Bernstein* polynomials as parametric basis; these polynomials, used in several application fields [105, 106], have been originally proposed in [107]. Given a maximum degree  $\bar{\ell}$ , the  $\ell$ -th order Bernstein polynomial  $b_{\ell, \bar{\ell}}$ , with  $\ell \leq \bar{\ell}$  is defined in the normalized domain  $x \in [0, 1]$  as

$$b_{\ell, \bar{\ell}}(x) = \binom{\bar{\ell}}{\ell} x^\ell (1-x)^{\bar{\ell}-\ell}, \quad \ell = 0, \dots, \bar{\ell}. \quad (5.13)$$

In our setting, polynomials expressed in the Bernstein basis are particularly attractive since they are uniformly positive in the domain  $[0, 1]$  and enable the use of constraints (5.12). We can thus define the multi-variate Bernstein basis function as

$$\xi_\ell(\boldsymbol{\vartheta}) = \prod_{i=1}^{\rho} \xi_{\ell_i}(\vartheta^i), \quad \xi_{\ell_i}(\vartheta^i) = b_{\ell_i, \bar{\ell}_i}(\vartheta^i) \quad (5.14)$$

### 5.2.2 Discussion

The introduction of the novel stability constraints (5.12), whose construction does not require any prior sampling of the parameter space, enables a dramatic reduction in the computational effort required to enforce uniform stability. In addition, if used simultaneously with positive semi-definite radial basis functions allow the generation of guaranteed stable high-dimensional parameterized macromodels. Numerical results in support of these statements are available in Section 5.3.

The only drawback of the proposed strategy is that constraints (5.12) are only sufficient conditions for the denominator's positive realness, which add another layer of conservativity to the method. With respect to the sampling-based approach, we therefore expect a deterioration of the final model-data accuracy, which we can not quantify a-priori. This problem will be analyzed in the following section, where we will compare the *Chebyshev* and *Bernstein* polynomial bases applying, respectively, the standard and the new set of constraints.

## 5.3 High-dimensional modeling – experiments

In this section, we will go through an experimental campaign aimed at assessing the performances of the proposed RBF modeling approach. In addition, we will compare the novel stability constraints (5.11) and (5.12) with the sampling-based approach.

### 5.3.1 Extensive performance comparison

Before testing the RBFs on high-dimensional examples, we report the results of an extensive testing campaign, originally presented in [101], aimed at

- comparing the modeling performances of the different parameterization schemes (both structured and unstructured), and
- demonstrate the advantages of using positive definite basis functions together with the proposed stability constraints (5.11) and (5.12).

In this regard, we considered a suite of 10 different benchmarks. All the considered devices are listed in Table 5.1, together with the associated parameters, their variation ranges and relevant literature references where additional details can be found. Structures 1 – 5 depend on  $\rho = 2$  independent parameters, while structures 6 – 10 depends on  $\rho = 3$  parameters. Among all the test-cases, we include both active devices, linearized around their operating point for small signal analysis, (structures 1 – 5) and passive interconnects (structures 6 – 10).

Despite the high-dimensional capabilities of the radial basis functions, we limit to three the number of parameters allowing the comparison with pre-existing approaches based upon multi-variate combinations of univariate polynomials (Chebychev, in our case) that are considered as reference.

For each structure in Table 5.1, we synthesized a uniformly stable macro-model according to the following parameterization schemes

- Structured
  1. multivariate Chebychev polynomials, for which uniform stability has been enforced according to the adaptive sampling scheme of Section 4.7 [4].
  2. multivariate Bernstein polynomial, defined in (5.14), where stability has been enforced with the constraints (5.12)
- Unstructured
  1. Gaussian RBF,
  2. Multiquadric RBF,
  3. Inverse multiquadric RBF.

For all the *unstructured* bases, uniform stability is enforced by exploiting the constraints (5.12).

In order to evaluate the extent of conservativity losses induced by constraints (5.12) with respect to the “sampling based” approach, we decided to provide a comparison with the Bernstein structured basis. This comparison is meaningful since, for a fixed maximum expansion degree, the polynomial approximation space is invariant with respect to change of basis. Thus, neglecting minor deviations induced by the different conditioning of the optimization problem (which are irrelevant for our purposes), the only measurable difference in model accuracy with respect to the reference is due to the different conservativity levels induced by the stability enforcement procedure.

In our comparative test, we evaluate the quality of the observed macromodeling techniques by means of the following metrics :

Table 5.1 Benchmark suite used for the comparative study of different parameterization schemes. For each test case 1–10 the table reports dataset information as: the number of parameters  $\rho$ , the total number of frequency samples  $\bar{k}$  for each of the  $\bar{m}$  available parametric responses, and details on structure type, individual parameters with their range, and references for additional info.

#	$\rho$	$\bar{k}$	$\bar{m}$	Type	Parameters	Range
1	2	273	341	Buffer [108]	Bias voltage (V) Temperature ( $^{\circ}C$ )	[0.5, 1.5] [20, 50]
2	2	293	231	Buffer [108]	Bias voltage (V) Temperature ( $^{\circ}C$ )	[0.5, 1.5] [20, 40]
3	2	213	119	LNA [108]	Bias voltage (V) Temperature ( $^{\circ}C$ )	[0.9, 1.2] [-30, 130]
4	2	235	35	LNA [108]	Bias voltage (V) Input Voltage (V)	[0.9, 1.2] [0.4, 0.6]
5	2	210	231	OpAmp [109]	Bias voltage (V) Gain	[1.1, 1.3] [1.01, 2]
6	3	831	935	OpAmp [109]	Bias voltage (V) Temperature ( $^{\circ}C$ ) Gain	[1.1, 1.3] [-30, 130] [1.01, 2]
7	3	901	300	TL network [110]	Inductance (nH) Inductance (nH) Capacitance (pF)	[4.5, 5.5] [9, 11] [0.9, 1.1]
8	3	901	300	TL network [110]	Inductance (nH) Capacitance (pF) Resistance ( $\Omega$ )	[9, 11] [0.45, 0.55] [45, 55]
9	3	901	300	TL network [110]	Inductance (nH) Resistance ( $\Omega$ ) Conductor width ( $\mu\text{m}$ )	[9, 11] [45, 55] [90, 210]
10	3	2000	216	TL filter [82]	Stub length (mm) Line length (mm) Load $\Gamma$	[6, 7] [9, 10] [0.1, 0.5]

- *Extraction Runtime*: the CPU time required to iteratively solve the PSK optimization problem, subject to the case-specific stability constraints. In order to ensure the comparison to be meaningful, we forced the number of PSK iterations to 10, which turned out to be sufficient to reach convergence in all the considered test-cases.
- *Model Complexity*: measures the complexity of the parameterized models based on the total number of free coefficients that are optimized during the model synthesis step.
- *Model Accuracy*: evaluates the model-data accuracy in terms of the RMS absolute error.

In order to better characterize the considered modeling strategies, we carried out two different experiments:

1. First, we derived a set of models by fixing the same *complexity* with the aim of comparing the attainable *accuracy* using the different parameterization strategies, together with the associated uniform stability constraints.
2. A second experiment has been performed, instead, by setting a fixed target model-data error  $\delta = 10^{-2}$ , that we consider to be an acceptable engineering accuracy for reduced macromodels. Hence, differently from the first experiment, the model accuracy has been finely tuned to reach the prescribed precision.

Following our work [101], the results of this section are obtained without resorting to advanced techniques to optimize the hyper-parameters of the RBFs. The number of RBFs has been selected by hand and their centers are located in correspondence to a random subset of the training data samples; the shape parameter  $\varepsilon$  is instead selected with a simple grid search.

The proposed results will show that, even with such a naive approach, the model-data accuracy is excellent.

**Remark 14.** *The shape parameter optimization step is extremely demanding, as it requires the repeated extraction of macromodels for several values of  $\varepsilon$ ; the time required for the optimization of the shape parameter is not considered*

in the *Extraction Runtime metric*. In Chapter 6 we are going to introduce optimized techniques that dramatically reduce the computational effort required to determine an optimal shape parameter.

### First experiment: fixed model complexity

The results of the first experiment are shown in panels a) and c) of Figure 5.3, for the two and three parameter cases, respectively. The complexity has been determined so that all the parameterization schemes could attain an acceptable accuracy; only three structures (3,9,10) resulted in slightly larger errors with multiquadric and inverse multiquadric RBFs.

First, let us compare the Chebychev and Bernstein polynomial bases. As expected, the use of Chebychev basis provides the most accurate models (see subpanels a.1 and c.1 of Figure 5.3), due to the reduced conservativity of the stability constraints based on adaptive samplings. On the other hand, this increased accuracy comes at the cost of an increased extraction runtime (mostly employed to adaptively determine the constraints locations), which becomes impractical for the three-dimensional test-cases, as evident in subpanel c.3. In this latter case, the speed-up is almost  $100\times$  for all the considered examples.

Considering now the RBFs parameterizations, we notice that the *multi-quadric* basis performs worst, while the *Gaussian* and the *inverse multi-quadric* are almost comparable. Compared, instead, with the polynomial bases, we notice the expected degradation with respect to Chebychev parameterization, but even with respect to the Bernstein basis with which shares the stability constraints. This may be the result of non-optimal estimates of the hyper-parameters; however, the accuracies are still acceptable.

In regard to the computational aspect, the extraction runtime of the RBF models is almost equal to the Bernstein case since, for any number of model coefficients, the complexity of the associated optimization problem is identical.

To summarize:

- The novel stability constraints (5.11) introduce a degradation on the model accuracy. However, the model-data accuracy is still acceptable for engineering purposes.



- The accuracy of the RBF models is, in general, worse than its counterpart based on Bernstein polynomials.
- The use of the new constraints (5.11) provides a significant speed-up (approximately,  $100\times$  for the 3-D models) with respect to the “standard” sampling based approach.

### Second experiment: fixed model accuracy

We now compare the modeling results at a fixed target model-data accuracy, which is set to  $\delta = 10^{-2}$  in terms of RMS worst-case absolute error. The model parametric orders are tuned accordingly.

Also this second set of experiments confirm that using positive semi-definite basis the extraction runtimes are dramatically reduced if compared with Chebyshev polynomials (see subpanels b.3 and d.3 of Figure 5.3). In addition, it is possible to appreciate the high-dimensional approximation capabilities of the RBFs by looking at subpanels b.2 and especially d.2, where it is evident the reduced complexity of the RBF models with respect to Chebyshev and Bernstein parameterizations. Here, the improvement is still moderate due to the small  $\rho$ ; see the next Section for high-dimensional examples where the RBF parameterization proves to be the only viable choice.

### 5.3.2 Scalability comparison

This second experimental campaign aims at showing the limits of “classical” approaches based on multi-variate combinations of polynomials in high-dimensional setting and, at the same time, demonstrating the excellent scalability properties of the RBFs.

We perform a comparative study considering lumped-distributed active circuit blocks. In particular, we consider a Low Noise Amplifier (LNA) shown in Figure 5.4 originally presented in [111], and we are interested to derive a linearized parameterized macromodel that reproduces the (linearized) small-signal input/output behavior at the ports “RF in” and “RF out”. The structure is parameterized by ten independent parameters (see Table 5.2 for the complete list): six are associated to lumped components (parasitic series inductances and

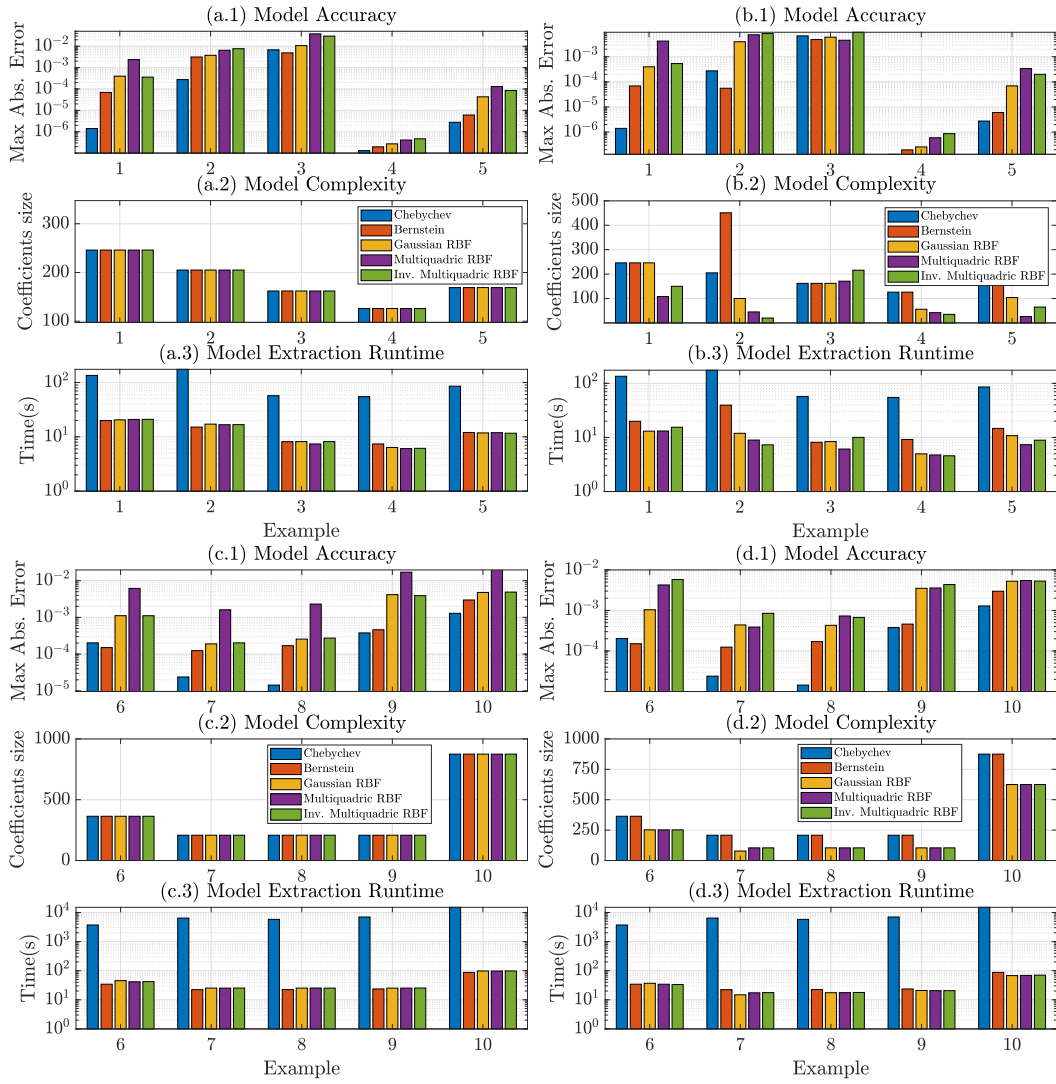


Fig. 5.3 Performance comparison of different parameterizations for ten benchmark structures: cases 1–5 (panels a, b) with  $\rho = 2$  parameters, and 6–10 (panels c, d) with  $\rho = 3$  parameters. Subpanels (a.*i*) and (c.*i*) for  $i = 1, 2, 3$  compare performance in terms of model accuracy, complexity, and runtime with a fixed total number of model coefficients for each individual example. Subpanels (b.*i*) and (d.*i*) show results obtained by tuning model complexity so that a prescribed model accuracy is attained. From [101] © 2020 IEEE

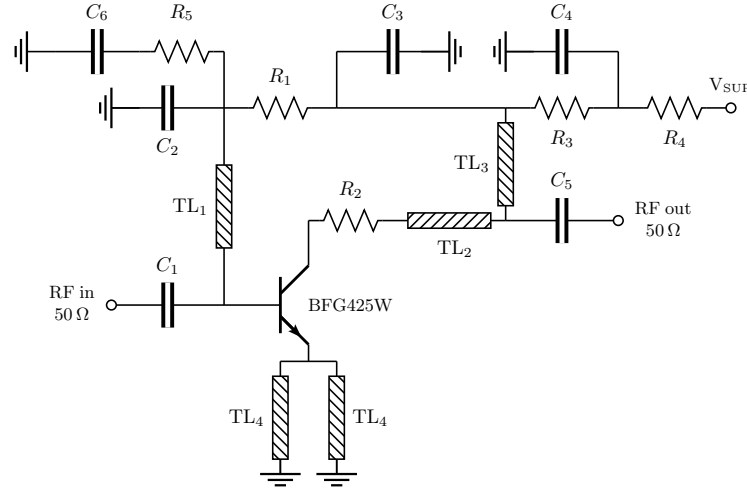


Fig. 5.4 A Low Noise Amplifier circuit [111]. From [101] © 2020 IEEE

shunt capacitances of the BJT), while the other four are geometrical properties of the transmission lines.

Table 5.2 LNA parameters. First six parameters: parasitic inductances and capacitances of the transistor. The remaining parameters are substrate thickness, conductor thickness, conductor width for lines  $TL_1$ ,  $TL_2$ ,  $TL_3$  and conductor width for line  $TL_4$ .

#	Parameter $\vartheta^i$	$\vartheta_{\min}^i$	$\vartheta_{\max}^i$
1	$L_b$ (nH)	0.88	1.32
2	$L_c$ (nH)	0.88	1.32
3	$L_e$ (nH)	0.20	0.30
4	$C_{cb}$ (pF)	0.0016	0.0024
5	$C_{be}$ (pF)	0.064	0.096
6	$C_{ce}$ (pF)	0.064	0.096
7	$h$ (mm)	0.45	0.55
8	$t_k$ ( $\mu\text{m}$ )	1.8	2.2
9	$w_{1,2,3}$ (mm)	0.225	0.275
10	$w_4$ (mm)	0.72	0.88

Adding one parameter at a time from  $\rho = 1$  to  $\rho = 10$  from the one listed in Table 5.2, we compute a set of  $M_\rho = 200\rho$  parameterized scattering responses through repeated HSPICE [112] runs. The  $M_\rho$  parametric samples are distributed on  $\Theta$  according to a Latin Hypercube distribution [113], that uniformly fills the available space, which makes it an excellent choice for our

purposes. Note that a uniform Cartesian sampling of high dimensional spaces would have resulted in an exponentially increasing number of responses, which of course is impractical.

For each of these ten (increasingly larger) dataset, we construct a uniformly stable model using both the polynomial Bernstein and the unstructured Gaussian bases. The rationale behind this choice is twofold:

1. we need to compare parameterizations for which the uniform stability constraints (5.12) are applicable, and
2. among the considered RBFs, the Gaussian appeared to provide the best performances according to the results of the previous comparative test.

All the models were built with a number of poles  $\bar{n} = 16$  and their complexity has been tuned to reach the prescribed RMS absolute error of  $\delta = 0.01$ .

In Figure 5.5 we report the results of the comparative test. As already highlighted in the comparative test, the unstructured RBF models suffer from a slight accuracy degradation, although the attained accuracy is higher than the considered threshold  $\delta = 0.01$ .

Considering the scalability properties, using the Bernstein polynomial basis, we are allowed to generate macromodels up to  $\rho = 6$ , due to an exponentially growing complexity that limits the applicability to larger parameter spaces. Conversely, in this high-dimensional setting, the superiority of the Gaussian RBF parameterization is evident from the bottom panel of Figure 5.5, which demonstrates a controlled growth of the model complexity as  $\rho$  increases. This proves that a mesh-free (positive semi-definite) RBF approach is ideal to synthesize guaranteed stable high-dimensional models.

In this regard, we go through the modeling steps for the largest 10-dimensional LNA amplifier. Based on the aforementioned rules, the available dataset is composed of  $\bar{m} = 2000$  parametric frequency responses (distributed according to a Latin Hypercube sampling), each made of  $\bar{k} = 701$  linearly spaced samples in the range 1 – 10 GHz. We remark that, among the 2000 available samples, only  $\bar{m}_t = 160$  were exploited to train the model, while the other were left for validation purposes. For such a high-dimensional modeling task, only RBF-based parameterization are viable solutions to obtain a

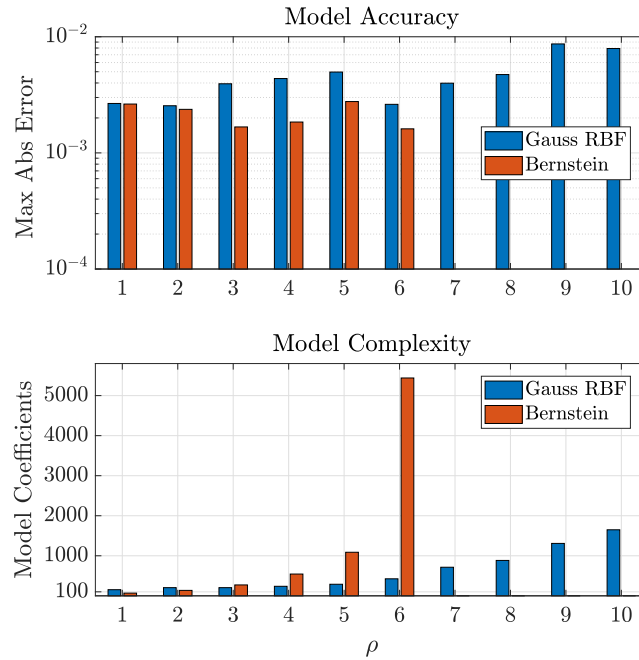


Fig. 5.5 Model accuracy and complexity of the LNA example obtained using Bernstein polynomials and Gaussian RBFs, for increasing parameter space dimension  $\rho$ .

uniformly stable macromodel. The documented results refer to the Gaussian RBF.

Following a grid search optimization, a sub-optimal shape parameter value has been found equal to  $\varepsilon = 0.03$ . The parameterized model has  $\bar{n} = 16$  poles, pre-computed (in correspondence of the centroid of the parameter space) by VF runs in the initialization step, and the parameterization is induced by  $\bar{\ell}_N = 90$  and  $\bar{\ell}_D = 5$  Gaussian basis for numerator and denominator, respectively.

The model extraction required only 91 seconds, and its accuracy with respect to the available validation samples is  $7.72 \cdot 10^{-3}$  (which is remarkable given the reduced number of coefficients compared to the parameter space dimension) that is confirmed in Fig. 5.6, that compares the model responses over a randomly selected subset of validation samples.

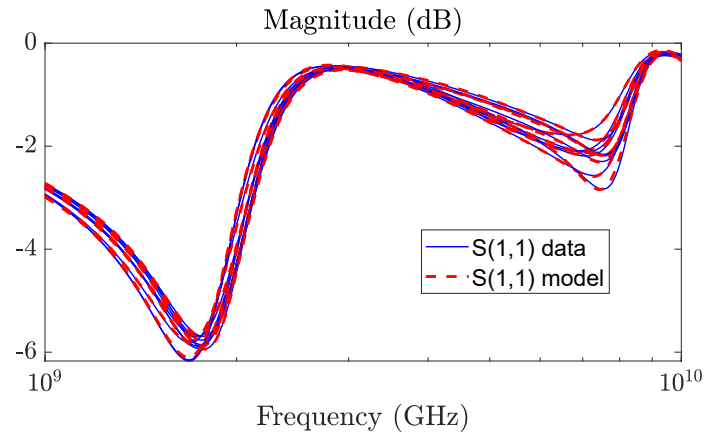


Fig. 5.6 Comparison of LNA model responses to corresponding raw validation responses, randomly selected in the 10-dimensional parameter space. From [101] © 2020 IEEE

## 5.4 Discussion

In this chapter, we introduced the topic of high-dimensional macromodeling. We argued that a classical approach based on multi-linear combinations of univariate polynomials is not feasible in case the number of parameters exceeds few units.

We addressed this problem following a mesh-free/unstructured approach that, as pointed out in the introductory section, is the only feasible way to address high-dimensional problems. In particular, we adopted Radial Basis Functions, and we performed a thorough comparison over several test cases

One of the most compelling results of this chapter is the formulation of uniform stability constraints that, at the cost of a slightly increased conservativity, allow the construction of guaranteed stable parameterized models, independently of the number of parameters. This was not possible with earlier approaches that suffer from the curse of dimensionality. We remark that even in low-dimensional settings, the use of these novel stability constraints lead to a major performance improvement in terms of computational efficiency.

In order to fully leverage the potential of RBFs it is required to finely tune their hyper-parameters. Although a grid search optimization may be successful (as demonstrated through numerical experiments), the computational effort is excessive.

---

In the following Chapter, we introduce several innovative and optimized strategies to select sub-optimal RBFs' hyperparameters. These will be used to develop a fully automated high-dimensional macromodeling algorithm, which may be considered as the main result of this work.

# Chapter 6

## Fully automated macromodeling

In this Chapter, we introduce several strategies for the automated selection of RBFs' hyper-parameters. The result is a complete set of algorithmic methods that autonomously determine the model orders and the RBFs' shape parameters. These are then used to develop a (high-dimensional) fully automated macromodeling strategy.

### 6.1 Hyper-parameters optimization

The adoption of unstructured Radial Basis Functions has proven to be a winning solution in the context of high-dimensional macromodeling. Besides their undisputable advantages in high-dimensional settings, the success of RBF methods strongly depends on some *hyper-parameters*, that encode the “geometrical” properties of the bases. Thus, the RBF hyper-parameters must be finely tuned to reach the desired model's accuracy and complexity. In Chapter 5 we voluntarily avoided discussing on the choice of hyper-parameter, whose optimization will be covered here.

In the following, we will consider only the Gaussian RBFs that, as already pointed out in Section 5.1.4, are defined upon two hyperparameters: location and number of RBFs (which we collectively consider as a unique set of hyper-parameters) and their shape parameter. In the following, we assume that the



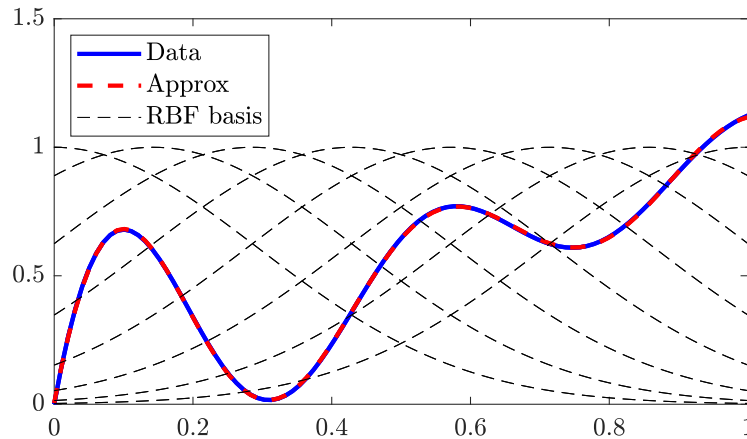


Fig. 6.1 Approximation results with finely tuned hyper-parameters.

shape parameter is shared among all the considered RBFs'. Studies on spatially variable shape parameters can be found in [114, 115].

In order to illustrate the importance of an accurate selection of the hyper-parameters, we consider a simple illustrative example. Let us assume that we aim at approximating a univariate function  $\check{s}(x)$ , represented with a blue solid line in Figure 6.2, defined as

$$\check{s}(x) = x + \frac{\sin(14x)}{1 + 7x} \quad (6.1)$$

in the domain  $x \in [0, 1]$ . Based on a set of 100 linearly spaced instances of  $\check{s}(x)$ , we first build an RBF *least-squares* approximation. The hyper-parameters have been optimized with a fine grid search optimization, that resulted in 10 RBFs (represented with black dashed lines in Figure 6.2) uniformly spaced in the  $[0, 1]$  domain and with a near-optimal<sup>1</sup> shape parameter  $\varepsilon \approx 2$ . As depicted in Figure 6.1, the resulting RBF approximation (red-dashed line) perfectly fits the data.

Then, starting from these near-optimal hyper-parameter estimates, we purposely change their values to better understand the effect on the approximation quality. In the left panel, we modify *only* the shape parameter while, on the right panel, we freeze the shape parameter and modify both the location and the number of RBFs. Let us analyze these results separately.

<sup>1</sup>We can not claim optimality, since our hyper-parameter optimization is based on a finite set of candidate configurations.

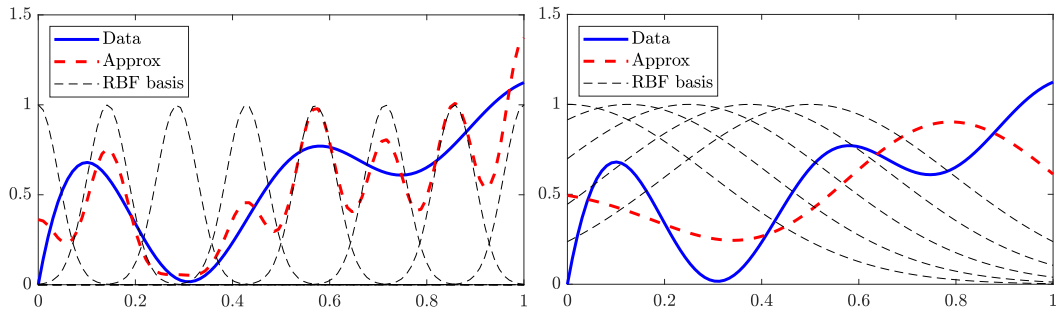


Fig. 6.2 Comparing the effect of the hyper-parameters on RBF approximation accuracy. Left panel: the number of RBF and their centers are optimized, while the shape parameter is too large; right panel: the shape parameter is optimized, while the distribution and the number of RBF is modified (see text).

- *Changing the shape parameter:* In this example, we increased  $\varepsilon$  up to 10, which caused the RBFs to narrow around their centers. This led to a major degradation of the approximation, that now exhibits a more “oscillatory” behavior induced by the narrower basis functions. We can conclude that, if the shape parameter exceeds a predefined (case-dependent) value, the approximation quality inevitably decreases. On the contrary, good approximation properties can be obtained for small  $\varepsilon$ . We will discuss on this subject in the dedicated Section 6.2.1.
- *Changing position and number of RBFs:* In this second case, we first reduce the number of RBF to 5, then we shift their distribution to the left-hand-side of the domain. Many numerical tests demonstrate that the accuracy does not depend significantly on the actual selection of the RBFs’ location, but rather is strongly conditioned by their number; we must thus ensure a sufficient number of basis functions to properly catch the target function variability.

On the other hand, although an excessively large number of RBFs does not necessarily lead to an accuracy degradation (except for possible numerical issues), it should be kept as small as possible to keep the model complexity under control.

Except for few simple cases, the optimization of the hyper-parameters can not be performed in a simple closed form and, in general, we must settle for sub-optimal approximated estimates whose computation is often delegated to black-box optimization algorithms [15]

Tuning the RBFs' hyper-parameters can be regarded as a supervised optimization problem that, based on sampled instances of the unknown target function, searches the most suitable hyper-parameter configuration by maximizing some performance metrics. Methods based on simple grid search techniques, or more sophisticated strategies based on maximum likelihood estimators [12, 15], *Bayesian* optimization and evolutionary algorithms [15] are routinely applied [13–15].

Unfortunately, a direct application of these strategies in our framework is not feasible, since:

- we *can not* be given with a unique set of sampled instances of the target function to approximate. In fact, for each different frequency value, the parametric dependence that we are required to approximate may undergo significant variations (e.g., parameterized resonances).
- All the most common techniques have been proposed in the context of linear models; instead, we are considering a rational RBF model, for which these strategies do not directly apply. Very few results regarding rational RBF approximation are available in literature [116, 117], but none addresses the optimization of the shape parameter.
- The evaluation of performance metrics is based upon the repeated extraction of intermediate models that, in our setting, may require unacceptably long runtimes; indeed, we remark that the synthesis of a parameterized macromodel requires the iterative solution of linearly re-weighted (possibly large) least-square problems.

We can conclude that, without an efficient scheme that allows a fast estimation of the RBFs' hyper-parameters, the use RBFs in high-dimensional macromodeling remains impractical. In this Chapter we aim at presenting a comprehensive set of techniques, originally introduced by the Author in [103, 118, 119] that optimize the evaluation of sub-optimal hyper-parameter values. In details, we will present a very efficient strategy for the optimization of the shape parameter that, interestingly enough, does not require any information about the target function. Then, we will propose two greedy approaches to select the number and locations of the RBFs.

**Remark 15.** *Even though the proposed hyper-parameter optimization schemes are presented as independent, they are in fact strongly correlated and should be interpreted as parts of a more general optimization strategy (see Section 6.4.2).*

## 6.2 A sub-optimal shape parameter selection strategy

We start investigating the problem of finding a (sub)-optimal value for the shape parameter  $\varepsilon$ . In the literature, this question has been tackled under numerous standpoints. On one hand, the problem has been addressed following more theoretical-grounded strategies; for instance, the results in [120] provide meaningful approximation error bounds based on  $\varepsilon$  and on the data sample distribution. Others, instead, target this subject following a “statistical” approach, suggesting to evaluate the shape parameter based on the optimization of some error indicators [12]. The former, although theoretically relevant, turn out to be of little practical use in our context. The latter, that are routinely adopted in most of the machine learning schemes (see, for instance, [121]), require repeated model extractions, resulting to be of very limited use in our framework. Thus, we aim at something different.

In the following, we will introduce a heuristic strategy, first presented in [118] then theoretically proved in [103], that estimates a “good” value of the shape parameter based only on the spectral properties of the PSK matrices, without the need of any information about the function to approximate.

### 6.2.1 Accuracy, conditioning, and shape parameter

In order to better understand which characteristics must be reflected by a “good” estimate of the shape parameter, we investigate its connections with the model accuracy and the condition number of the PSK matrices. In support of the following observations, we make reference to Figure 6.3 that compares the

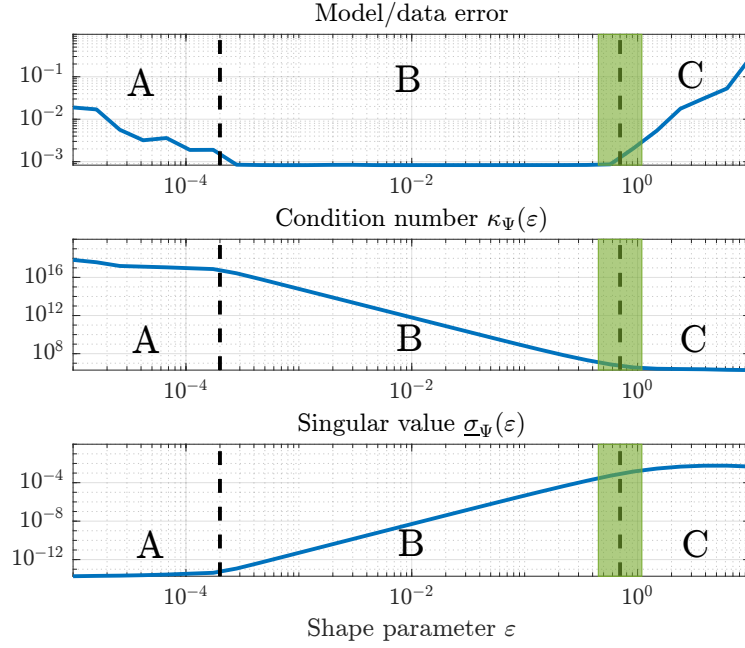


Fig. 6.3 Dependence on the shape parameter  $\varepsilon$  of model-data error (top panel), condition number of matrix  $\Psi(\varepsilon)$  (middle panel), and least singular value of matrix  $\Psi(\varepsilon)$  (bottom panel). For all panels, three regions are highlighted: *A* (numerical instabilities associated with small  $\varepsilon$ ), *B* (acceptable model accuracy), *C* (loss of approximation capabilities). Green region: candidate sub-optimal shape parameter values. From [103] © 2021 IEEE

above variables of interest in the case of a square integrated coil, parameterized by its side length.<sup>2</sup> Let us analyze the two opposing asymptotic behaviors:

1. *Case*  $\varepsilon \rightarrow \infty$ :

Based on the observations in Section 6.1, for increasing values of  $\varepsilon$  the Gaussian RBFs progressively narrow around their centers  $\mathring{\vartheta}_\ell$ . In the limit case of  $\varepsilon \rightarrow \infty$ , the RBF takes the form of a “needle”, centered in  $\mathring{\vartheta}_\ell$ . The asymptotic behavior is

$$\xi_\ell^\infty(\vartheta) = \begin{cases} 1 & \text{if } \vartheta = \mathring{\vartheta}_\ell \\ 0 & \text{if } \vartheta \neq \mathring{\vartheta}_\ell \end{cases} \quad (6.2)$$

<sup>2</sup>We remark that the results are general and apply independently on the considered test-case.

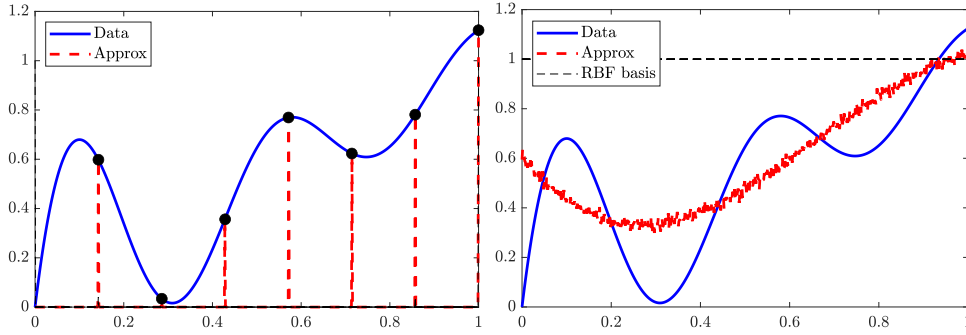


Fig. 6.4 Comparing the approximation performances for different values of the shape parameter. Left panel: large  $\varepsilon$ , representative of the asymptotic case  $\varepsilon \rightarrow \infty$ . Right panel: small  $\varepsilon$ , representative of the flat limit  $\varepsilon \rightarrow 0$ .

A model built in correspondence to  $\varepsilon \rightarrow \infty$  exactly interpolates the data at the RBF centers and identically vanishes elsewhere, as depicted in the left panel of Figure 6.4. This asymptotic condition always results in a poor modeling accuracy, since no continuous approximations can be achieved.

This is observable in the top panel of Figure 6.3 (region C) that, in the case of our running example, illustrates the growth of the model-data error attained for increasingly large values of  $\varepsilon$ .

By contrast, when  $\varepsilon \rightarrow 0$  the fitting problem is very well conditioned. In fact, based on (6.2), under rows and columns permutations, the kernel matrix  $\mathbf{K}_{N,D}(\varepsilon)$  becomes

$$\mathbf{K}_{N,D}(\infty) = \begin{pmatrix} \mathbf{I}_{\bar{\ell}} \\ \mathbf{0} \end{pmatrix} \quad (6.3)$$

that, by definition, has the optimal condition number  $\kappa_{\mathbf{K}}(\infty) = 1$ .

Thus, following the definitions in Section 5.2 it is readily understood that the good numerical properties of  $\mathbf{K}_{N,D}(\infty)$  are inherited by the whole set of PSK matrices. This is confirmed by the middle panel of Figure 6.3, in region C.

## 2. Case $\varepsilon \rightarrow 0$ :

When  $\varepsilon \rightarrow 0$  the Gaussian RBFs increasingly flatten, becoming somehow “insensitive” to their separating distances. In turn, the associated kernel

matrix becomes a matrix of all ones

$$\mathbf{K}_{N,D}(0) = \mathbf{1}_{\bar{m} \times \bar{\ell}} \quad (6.4)$$

which is singular and, thus, the approximation can not be computed numerically. Even for small (yet not vanishing) values of  $\varepsilon$  it is common to run into numerical issues (see Figure 6.4, right panel).

This behavior is illustrated in Figure 6.3 (middle panel, region B) where the condition number grows exponentially for decreasing values of  $\varepsilon$ . In addition, in correspondence to “region A” of the top panel, it is clear that the accuracy degradation is induced by a very poor conditioning.

On the other hand, a set of remarkable results surprisingly prove that the limit of the interpolation when  $\varepsilon \rightarrow 0$  indeed exists, and it has been proven to coincide with the De Boor/Ron “least polynomial” interpolant [122]. Then, we expect a polynomial accuracy in the so-called RBFs’ “flat limit”, which is particularly attractive when approximating smooth functions.

**Remark 16.** *The well-behaved polynomial “flat limit” has encouraged further investigations. In particular, under appropriate change of basis, the mentioned numerical issues can be greatly alleviated, see [123] and [124] for a review. However, better conditioned bases are always found in the form of multi-variate polynomials, that scale badly in high-dimensional spaces and can not be adopted in our schemes.*

We highlight that, although there are proven results for the asymptotic limit  $\varepsilon \rightarrow 0$ , good approximation results are still obtained with non-vanishing, yet small, shape parameters. This is demonstrated in the top-panel of Figure 6.3, showing that good approximation results can be obtained in the whole region B.

Based on the above observations, the “good” shape parameter value must be sought as sufficiently small  $\varepsilon$  to ensure accuracy, yet sufficiently large to prevent numerical problems. In the following, we further elaborate on the above statements.

## 6.2.2 On the numerical conditioning and accuracy

In the following, we will prove that the behavior of the condition number  $\kappa_{\Psi}(\varepsilon)$  illustrated in Figure 6.3 is typical for Gaussian RBF kernel matrices. We will use this results to set up the optimization algorithm in Section 6.2.3.

First, we recall that  $\kappa_{\Psi}(\varepsilon) = \bar{\sigma}_{\Psi}(\varepsilon)/\underline{\sigma}_{\Psi}(\varepsilon)$ , where  $\bar{\sigma}_{\Psi}(\varepsilon)$  and  $\underline{\sigma}_{\Psi}(\varepsilon)$  are, respectively, the leading and the least singular values of matrix  $\Psi(\varepsilon)$ . These singular values are separately analyzed below.

1. *The least singular value:*

For a complete characterization of the least singular value  $\underline{\sigma}_{\Psi}(\varepsilon)$ , an in-depth analysis of the spectral properties of PSK sub-matrices is required. This analysis is presented in Appendix B, where we prove the following main result.

**Theorem 6.** *Let  $\Psi^{\nu}(\varepsilon)$  be the iteration dependent PSK regressor defined in (5.6). Let us additionally assume that the associated kernel matrices  $\mathbf{K}_{N,D}(\varepsilon)$  are constructed so that  $\bar{m}^t > \bar{\ell}_{N,D} + 2$ . Then, if  $\varepsilon \rightarrow 0$  the least singular value  $\underline{\sigma}_{\Psi}(\varepsilon)$  decays asymptotically as*

$$\underline{\sigma}_{\Psi}(\varepsilon) = O(\varepsilon^{\tau}), \quad \tau \geq 4, \quad \tau \in \mathbb{N}. \quad (6.5)$$

This statement is confirmed by the bottom panel of Figure 6.3, region B, where the least singular value  $\underline{\sigma}_{\Psi}(\varepsilon)$  decays linearly with a constant  $\tau \in \mathbb{N}$ , when plotted on a log-log scale. We highlight that the flattening in region A is induced only by truncation errors when evaluating the SVD.

2. *The leading singular value:*

Conversely, it can be proven that the leading singular value  $\bar{\sigma}_{\Psi}(\varepsilon)$  does not to change significantly as  $\varepsilon$  changes. In particular, we have the following result,

**Proposition 1.** *Let  $\mathbf{K}(\varepsilon)$  be a Gaussian Kernel Matrix as defined in (5.9). Independently of the value  $\varepsilon$ , its leading singular value  $\bar{\sigma}_{\mathbf{K}}(\varepsilon)$  is bounded in the interval  $[1, \sqrt{\bar{m}^t \bar{\ell}}]$ .*



The proof is reported in Appendix C.

In particular, the leading singular value is analytically known to be exactly 1 and  $\sqrt{\bar{m}^t \bar{\ell}}$ , for  $\varepsilon \rightarrow 0$  and  $\varepsilon \rightarrow \infty$ , respectively. Thus, based on the construction of the PSK regressor detailed in Section 5.2 we conclude that the leading singular value  $\bar{\sigma}_{\Psi}(\varepsilon)$  of  $\Psi(\varepsilon)$  attains values in a closed interval that does not depend on  $\varepsilon$ . More precisely,

$$\bar{\sigma}_{\Psi}(\varepsilon) \sim C \quad (6.6)$$

where  $C$  is a positive constant.

By assembling the above observations, we can conclude that

$$\kappa_{\Psi}(\varepsilon) = \frac{\bar{\sigma}_{\Psi}(\varepsilon)}{\underline{\sigma}_{\Psi}(\varepsilon)} \sim \frac{C}{\underline{\sigma}_{\Psi}(\varepsilon)} \quad \varepsilon \rightarrow 0 \quad (6.7)$$

where, by virtue of Theorem 6, it holds that  $\underline{\sigma}_{\Psi}(\varepsilon) = O(\varepsilon^\tau)$ . We thus finally conclude that the condition number  $\kappa_{\Psi}(\varepsilon)$  grows as  $O(1/\varepsilon^\tau)$  for  $\varepsilon \rightarrow 0$ , as demonstrated numerically in Figure 6.3, region B. In addition, the middle and bottom panel confirm that the least singular value is almost inversely proportional to the condition number within the region B.

To summarize:

1. large values of  $\varepsilon$  lead to very well-conditioned fitting problems, but the associated models are inaccurate since the RBFs are too narrow;
2. small values of  $\varepsilon$  would produce accurate models, but the fitting problem is prone to numerical instabilities.

Shape parameter values belonging to center region B in Figure 6.3 concurrently fulfill the above conflicting requirements, since

1. they are small enough to guarantee a proper model parameterization;
2. they are sufficiently large to ensure a reasonable conditioning of the PSK fitting problem.

In order to guarantee the best possible condition number, we want to pick a sub-optimal value of  $\varepsilon$  that minimizes  $\kappa_{\Psi}(\varepsilon)$ . Thus, instead of considering the whole region B, we narrow our search to the *green-shaded* region, located at the interface of areas B and C.

### 6.2.3 An algorithm for choosing the shape parameter

Developing on the above considerations, we can introduce our efficient algorithm for selecting a sub-optimal value of the shape parameter  $\varepsilon^*$ . As anticipated, we aim at detecting the green-shaded area illustrated in Figure 6.3, located at the interface of regions B and C.

Exploiting the least singular value as a proxy for the condition number, we are thus interested in pinpointing the corner point between the scaling range of  $\underline{\sigma}_{\Psi}(\varepsilon)$  as  $O(\varepsilon^\tau)$ , and the subsequent nearly constant range.

Following the strategy proposed in [103, 118], we pre-compute a set of shape parameter-singular value pairs (red filled dots in Figure 6.5)

$$\{(\varepsilon_t, \underline{\sigma}_{\Psi}(\varepsilon_t)), t = 1, \dots, \bar{t}\}$$

over a set  $\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{\bar{t}}\}$  of increasing logarithmically spaced shape parameter candidates (3–4 points per decade have proved to be sufficient). We consequently define  $\epsilon_t = \log \varepsilon_t$  and  $\varrho_t = \log \underline{\sigma}_{\Psi}(\varepsilon_t)$ .

In order to localize the green-shaded corner point, we benefit from the result of Theorem 7 that ensures a power-law decay for  $\underline{\sigma}_{\Psi}(\varepsilon)$  for  $\varepsilon \rightarrow 0$ . Our proposed algorithm works iteratively; at each iteration  $J \leq \bar{t}$ , we construct a log-log regression line  $\zeta_J(\varepsilon) = \alpha_J \log \varepsilon + \beta_J$  (black thick line in Figure 6.5) on the current pairs  $\{(\epsilon_t, \varrho_t), t = 1, \dots, J\}$ . The corner point of our interest is assumed to be detected when the relative deviation  $\Delta_{J+1} = |\zeta_J(\varepsilon_{J+1}) - \varrho_{J+1}| / \varrho_{J+1}$  exceeds a predefined threshold  $\Delta_{th}$ . Therefore, assuming that at iteration  $J^*$  we have  $\Delta_{J^*+1} > \Delta_{th}$ , the optimization procedure stops, and we set  $\varepsilon^* = \varepsilon_{J^*}$ .

The following Algorithm 2 provides a pseudo-code for the above procedure.

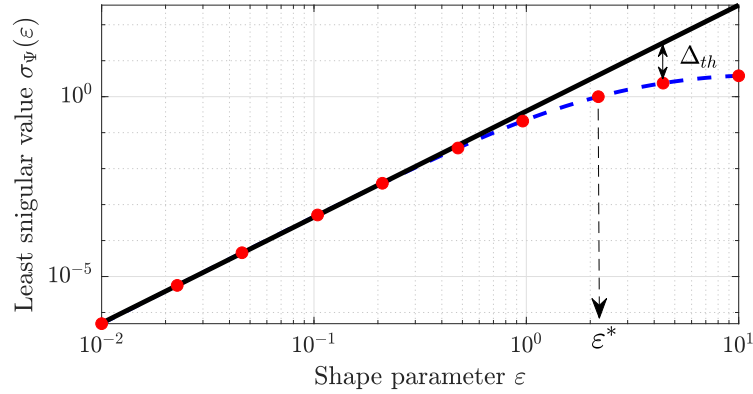


Fig. 6.5 Iterative regression (black thick line) on shape parameter-singular value pairs  $\{\epsilon_t, \rho_t\}$  (red dots). From [103] © 2021 IEEE

### 6.2.4 From regressor to kernel matrices

The procedure devised in Algorithm 2 would require iterated estimations of  $\epsilon$ , one at each PSK iteration that, in turn, would demand for expensive repeated (least) singular value computations. The following observations highlight that, fortunately, this is not necessary, since

- Even if the regressor matrix  $\Psi^\nu(\epsilon)$  is updated at each PSK iteration  $\nu$ , we can be convinced that the dependence of its least singular value on  $\epsilon$  is nearly invariant through iterations. Therefore, the sub-optimal shape parameter value  $\epsilon^*$  can be estimated once at the first PSK iteration, and reused at the subsequent iterations.
- The dependence on  $\epsilon$  of the least singular value  $\sigma_{\Psi}(\epsilon)$  of  $\Psi^\mu(\epsilon)$  is inherited by the Fast-PSK matrix  $\bar{\mathbf{R}}^\mu(\epsilon)$  (see Section 2.3.3 and [103]), which is significantly smaller. Sub-optimal shape parameter values can be thus inferred based on the singular values of  $\bar{\mathbf{R}}^\mu(\epsilon)$  in place of  $\Psi^\mu(\epsilon)$ , enabling a considerable performance improvement.
- A substantially more efficient implementation relies on the RBF kernel matrices  $\mathbf{K}_N(\epsilon)$ ,  $\mathbf{K}_D(\epsilon)$  associated, respectively, to the model numerator and denominator. By construction, the kernel matrices are much smaller than both the full and reduced PSK regressors  $\Psi^\mu(\epsilon)$  and  $\bar{\mathbf{R}}^\mu(\epsilon)$ , enabling for an extremely fast computation of their singular values. In Appendix B we will prove that the dependence on  $\epsilon$  of  $\Psi^\mu(\epsilon)$  (or, equivalently  $\bar{\mathbf{R}}^\mu(\epsilon)$ )

---

**Algorithm 2** Sub-optimal shape parameter selection [118]

---

**Require:** Log-spaced samples  $\{\varepsilon_1, \dots, \varepsilon_{\bar{t}}\}$ .  
**Require:** Stopping threshold  $\Delta_{th}$   
**Require:** Reference matrix  $\mathbf{Z}(\varepsilon)$ :  $\Psi(\varepsilon)$ ,  $\bar{\mathbf{R}}(\varepsilon)$ , or  $\mathbf{K}_{N,D}(\varepsilon)$

- 1: Compute  $\epsilon_t = \log \varepsilon_t$  and  $\varrho_t = \log \underline{\sigma}_{\mathbf{Z}}(\varepsilon_t)$  for  $t = 1, \dots, \bar{t}$
- 2: **for**  $J = 2, \dots, \bar{t}$  **do**
- 3:     Build regression line  $\zeta_J(\varepsilon) = \alpha_J \log \varepsilon + \beta_J$  using pairs  $\{(\epsilon_t, \varrho_t), t = 1, \dots, J\}$
- 4:     Evaluate  $\Delta_{J+1} = |\zeta_J(\varepsilon_{J+1}) - \varrho_{J+1}| / \varrho_{J+1}$
- 5:     **if**  $\Delta_{J+1} > \Delta_{th}$  **then**
- 6:         Set  $\varepsilon^* = \varepsilon_J$
- 7:         **break**
- 8:     **end if**
- 9: **end for**
- 10: **return** Sub-optimal shape parameter  $\varepsilon^*$ .

---

is shared with  $\underline{\sigma}_{\mathbf{K}_N}(\varepsilon)$  and  $\underline{\sigma}_{\mathbf{K}_D}(\varepsilon)$ ; therefore, sub-optimal shape parameter estimates can be efficiently recovered exploiting the reduced kernel matrices. In addition, this last implementation enables for an individual selection of  $\varepsilon$  for numerator and denominator, separately; this is not possible in the previous approaches.

In summary, we propose here an efficient algorithm that estimates sub-optimal numerator and denominator shape parameter values  $\varepsilon_N^*$  and  $\varepsilon_D^*$ , applying Algorithm 2 to the kernel matrices  $\mathbf{K}_N(\varepsilon)$  and  $\mathbf{K}_D(\varepsilon)$ , separately.

### 6.2.5 Numerical experiment

In order to assess its performances, we will compare the proposed strategy with a standard grid search. We want to construct a RBF model for a double folded microstrip filter, introduced in Section 2.3.3. This structure depends only on one external parameter, namely the length of the microstrip stubs. Although it may seem a quite simplistic example for our RBF high-dimensional framework, we remark that the results are general and still hold in higher dimensional spaces; our choice for a reduced dimensional test case stems from the need for a reasonably fast grid search to use as reference benchmark. We assume that the sub-optimal shape parameter value must be sought in the interval

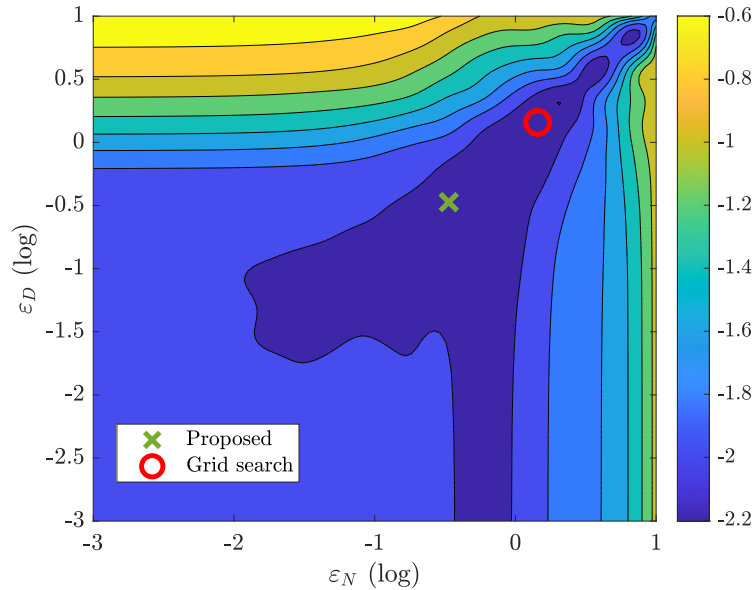


Fig. 6.6 Testing the performances of the proposed shape parameter selection algorithm. We illustrate with a suitable color-scale the model-data error obtained spanning  $\varepsilon_N$  and  $\varepsilon_D$  over a uniform grid. The result of our approach (green cross) is compared with a reference grid search (red circle).

$[10^{-3}, 10]$  (both for numerator and denominator), that is discretized in  $\bar{t} = 20$  logarithmically spaced samples  $\varepsilon_t$ . The selection of the stopping threshold  $\Delta_{th}$  is fundamental to ensure good performance from the proposed estimation algorithm. Unfortunately, we do not still have theoretically grounded rules to guide a case-dependent selection of the stopping threshold. Nonetheless, in all the considered test-cases (more than the ones reported in this manuscript) we noticed that accurate results can be obtained setting  $\Delta_{th}$  to  $0.03 - 0.04$ ; in all the numerical examples we will propose in the following, we will use  $\Delta_{th} = 0.04$ . Still, further studies are in order to gain a better understanding.

Based on the above settings, we ran our algorithm and compared the result with a reference grid search performed on a uniform  $20 \times 20$  grid spanning the same range  $[10^{-3}, 10]$ . Figure 6.6 illustrates the outcome of our comparative test. Even if the solution found with Algorithm 2 (green cross) differs from the optimal solution (red circle), both correspond to fairly accurate models; in fact, the grid search results in a model-data worst case error of  $6.8 \cdot 10^{-3}$ , while the proposed heuristic method produce a slightly less accurate model, whose error is  $8.6 \cdot 10^{-3}$ .

Another key aspect is the computational effort: the grid search took 270 seconds to fully explore the  $20 \times 20$  grid, while the proposed strategy required only 0.16 seconds to compute a sub-optimal estimate. This means that our strategy is approximately 1700 times faster, at the cost of an irrelevant degradation of the model accuracy with respect to the global optimum.

**Remark 17.** *We are aware that there exist more advanced techniques to estimate more efficiently the shape parameter [12–15]. However, all of these methods require, at least, the extraction of one macromodel; the proposed strategy does not.*

*As a comparison, one model extraction (without shape parameter estimation) requires approximately 0.8 seconds, which is 5 times slower than what required by our algorithm for a full  $\varepsilon$  estimation.*

*We conclude that any method requiring the repeated extraction of intermediate models can not be computationally more efficient than our proposed strategy.*

### 6.3 Choosing RBFs' location and number

We now turn to considering the other hyper-parameters: the RBFs' locations and their number (also referred to as *parametric order*). Although they can be considered as distinct, their close correlation necessitates a concurrent optimization; in the following, we will thus treat these two as a single hyper-parameter set.

**Remark 18.** *We recall that all the hyper-parameters optimization steps are distinct, but interdependent; in the following we will thus assume the shape parameter to be optimized at some optimal (or, sub-optimal) value  $\varepsilon^*$  (obtained, for instance, with Algorithm 2).*

There exist well-established techniques that optimize both the positioning and the number of RBFs. The most renown (thanks to their high accuracy and implementation simplicity) are based on data splitting, such as *k-fold* or *leave-one-out* cross validation (see, for instance, [125, 126]). The interested Reader can find more advanced techniques in [127–129]. Despite all of these

strategies are routinely used to solve machine learning tasks, their successful adoption requires the repeated extractions of numerous intermediate models, which is unpractical in our setting.

Others, instead, proposed some techniques built upon clustering algorithms that optimize the RBFs locations based on the geometric distribution of the training data samples space [128, 130, 131]. In our setting, however, the particular distribution of the data samples in the input space is not necessarily representative of their variability.

Therefore, none of the most common schemes seems to be adequate for our parameterized macromodeling setting. In [119] and [103] we thus introduced two ad-hoc strategies:

1. the first concurrently optimizes both the model orders and the RBFs' centers;
2. the second, instead, assumes a predefined distribution for the centers, and only the model orders are optimized.

In the following, we will go through the details of both these approaches, for which we will provide relevant numerical examples and a final comparison.

### 6.3.1 Method 1: concurrent optimization

This first method, originally presented in [119], proposes a concurrent optimization of both the number of RBFs (model orders) and of their positioning on the parameter space following an iterative greedy strategy. The main idea can be briefly summarized as follows: at each iteration  $\nu$ , we add a new RBF whose center is positioned in correspondence to a (sub-optimal) location, in order to progressively reduce the model-data error through iterations; the iterations stop when the accuracy is sufficiently adequate.

Such a centers' optimization procedure can not be directly performed on the continuous parameter space, on which we therefore induce a prior discretization on  $M$  samples: the finer is the resolution, the most accurate is the selection of locally optimal centers. The discretization procedure amounts to determine  $M$  frequency responses evaluated in correspondence of as many

parameters combinations  $\boldsymbol{\vartheta}_m$ , that we collect in a (iteration *independent*) set  $\mathcal{M} = \{\boldsymbol{\vartheta}_1, \dots, \boldsymbol{\vartheta}_M\}$ .

This set is split in two mutually disjoint (iteration *dependent*) subsets  $\mathcal{M}_\nu^t$  and  $\mathcal{M}_\nu^v$ , composed of  $m_\nu^t$  and  $m_\nu^v$  samples, respectively. We have that  $\mathcal{M} = \mathcal{M}_\nu^t \cup \mathcal{M}_\nu^v$  and  $M = m_\nu^t + m_\nu^v$ . The parametric samples in set  $\mathcal{M}_\nu^t$  are used to train the model, while the samples in  $\mathcal{M}_\nu^v$  are used for validation purposes. A third set denoted as  $\mathcal{M}_\nu^c \subseteq \mathcal{M}_\nu^t$ , collects the locations of the  $m_\nu^c < m_\nu^t$  RBFs that are used to construct the model  $\mathbf{H}^\nu(s, \boldsymbol{\vartheta})$  at the current iteration  $\nu$ .

**Remark 19.** *Since we can not precisely ascribe the presence of large local model-data errors to the model's numerator or denominator, we assume that they share the same common set of basis functions.*

Let us now go through the algorithm steps.

1. *Initialization:* As anticipated, the set  $\mathcal{M}$  is not augmented through iterations (e.g., via on demand solver calls). In order to have a complete and meaningful representation of the parameter space, we must initialize  $\mathcal{M}$  in a way that the samples  $\boldsymbol{\vartheta}_m$  uniformly cover  $\Theta$ . To this end, it is possible to adopt low discrepancy sequences (such as Sobol and Halton sequences) or Latin Hypercube sampling schemes.

The set  $\mathcal{M}$  is then partitioned in the two mutually disjoint *training* and *validation* subsets. In order to control the model accuracy over the entire parameter space, also the training samples must be uniformly distributed. To this end, we consider a  $\rho$ -dimensional truncated Sobol sequence  $\mathcal{S}^t$  [132], composed of  $m_0^t$  samples

$$\mathcal{S}^t := \{\mathbf{s}_i : i = 1, \dots, m_0^t\} \quad (6.8)$$

The initial training set  $\mathcal{M}_0^t$  is then populated with the elements of  $\mathcal{M}$  that are the nearest neighbors to the elements of  $\mathcal{S}^t$ .

The same procedure is repeated to populate the set  $\mathcal{S}^c$  of RBF centers. We thus define another truncated Sobol sequence  $\mathcal{S}^c$ , composed of  $m_0^c < m_0^t$  samples, and the initial centers are chosen as the nearest neighbors of  $\mathcal{M}_0^t$  to  $\mathcal{S}^c$ .



**Remark 20.** *In order to ensure a sufficiently over-determined fitting problem that prevent possible overfitting issues, we enforce that  $m_\nu^t \geq 2m_\nu^c$  at all the greedy iterations.*

If these Sobol sequences are defined upon a common initialization seed (which is often the case), then  $\mathcal{S}^c \subseteq \mathcal{S}^t$ ; therefore, the initial RBF centers can be selected as the first  $m_0^c$  element of  $\mathcal{S}^t$  and there is no need of constructing the sequence  $\mathcal{S}^c$ . The first model  $\mathbf{H}^0(s, \boldsymbol{\vartheta})$  is thus initialized with (common) numerator and denominator parametric orders  $\bar{\ell}_N = \bar{\ell}_D = m_0^c$ .

2. *Adaptive augmentation* Assuming to be at the  $\nu$ -th iteration, we construct the model  $\mathbf{H}^\nu(s, \boldsymbol{\vartheta})$  based on the sets  $\mathcal{M}_\nu^t, \mathcal{M}_\nu^c, \mathcal{M}_\nu^v$  available from the previous iteration; we assume the shape parameter to be optimized with Algorithm 2, At this stage, we evaluate the worst-case RMS error  $E^\nu(\boldsymbol{\vartheta}_m)$ , defined as

$$E^\nu(\boldsymbol{\vartheta}_m) = \max_{i,j} \sqrt{\frac{1}{K} \sum_{k=1}^K \left| \mathbf{H}_{i,j}^\nu(j\omega_k, \boldsymbol{\vartheta}_m) - (\check{\mathbf{H}}_{k,m})_{i,j} \right|^2}, \quad (6.9)$$

over all the validation samples in  $\mathcal{M}_\nu^v$ . Based upon this metric, we select the new RBF center according to

$$\tilde{\boldsymbol{\vartheta}}^\nu = \arg \max_{\boldsymbol{\vartheta}_m \in \mathcal{M}_\nu^v} E^\nu(\boldsymbol{\vartheta}_m) \quad (6.10)$$

as the particular validation sample for which the model-data error is largest. The sets are then updated as

$$\begin{aligned} \mathcal{M}_{\nu+1}^t &= \mathcal{M}_\nu^t \cup \tilde{\boldsymbol{\vartheta}}^\nu \\ \mathcal{M}_{\nu+1}^c &= \mathcal{M}_\nu^c \cup \tilde{\boldsymbol{\vartheta}}^\nu \\ \mathcal{M}_{\nu+1}^v &= \mathcal{M}_\nu^v \setminus \tilde{\boldsymbol{\vartheta}}^\nu \end{aligned} \quad (6.11)$$

The algorithm stops at iteration  $\bar{\nu}$  when the following condition is reached,

$$\max_{\boldsymbol{\vartheta}_m \in \mathcal{M}_{\bar{\nu}-1}^v} E^{\bar{\nu}}(\boldsymbol{\vartheta}_m) < E_{\max} \quad (6.12)$$

where  $E_{\max}$  is a user-defined error threshold. If the above condition is never reached, the algorithm safely stops whenever  $\nu = \nu_{\max}$  in order to prevent uncontrolled behaviors.

The final model has parametric orders  $\bar{\ell}_N = \bar{\ell}_D = m_0^c + \bar{\nu}$ .

The above centers' selection greedy algorithm is summarized in form of pseudo-code in Algorithm 3

---

**Algorithm 3** Adaptive RBFs center selection

---

**Require:** Parametric dataset  $\mathcal{M}$

**Require:** Control parameters  $E_{\max}$ ,  $\nu_{\max}$ ,  $m_0^t$ ,  $m_0^c$

- 1: Populate  $\mathcal{M}_0^t$  and  $\mathcal{M}_0^c$
  - 2: Set  $\mathcal{M}_0^v = \mathcal{M} \setminus \mathcal{M}_0^t$
  - 3: Set  $\nu = 0$
  - 4: **repeat**
  - 5:     Optimize  $\varepsilon$  with Algorithm 2
  - 6:     Generate model  $\mathbf{H}^\nu(s, \boldsymbol{\vartheta})$  on training samples  $\mathcal{M}_\nu^t$
  - 7:     Evaluate error function  $E^\nu(\boldsymbol{\vartheta}_m)$ ,  $\forall \boldsymbol{\vartheta}_m \in \mathcal{M}_\nu^v$
  - 8:     Define new RBF center  $\tilde{\boldsymbol{\vartheta}}^\nu$  as in (6.10)
  - 9:     Update sets  $\mathcal{M}_\nu^t$ ,  $\mathcal{M}_\nu^v$ ,  $\mathcal{M}_\nu^c$  as in (6.11)
  - 10:    Update iteration index  $\nu = \nu + 1$
  - 11: **until**  $E^\nu(\boldsymbol{\vartheta}_m) < E_{\max}$ ,  $\forall \boldsymbol{\vartheta}_m \in \mathcal{M}_\nu^v$  or  $\nu = \nu_{\max}$
  - 12: **return** Final model  $\mathbf{H}^{\bar{\nu}}(s, \boldsymbol{\vartheta})$
- 

### Numerical experiment (1) - a linearized buffer model

We first study a 2-parameter example to better illustrate the proposed strategy. Considering a two-stage buffer (linearized around its working condition for small-signal operation, parameterized by the operation temperature  $T = [20, 50]^\circ\text{C}$  and the bias voltage  $V_{\text{dd}} \in [0.5, 1.5]$  V (see [108] for full details). The full dataset  $\mathcal{M}$  is composed of  $M = 341$  parametric frequency responses computed through small-signal AC sweeps in a circuit solver, arranged in a  $11 \times 31$  Cartesian grid. Each frequency response includes  $K = 274$  frequency samples in the band  $f_{\min} = 0$  Hz,  $f_{\max} = 10$  THz (to show the high-frequency behavior).

The proposed centers' selection algorithm is initialized with  $m_0^t = 40$ ,  $m_0^c = 10$ , the maximum allowed number of additional RBFs is  $\nu_{\max} = 30$ , and the error threshold  $\mathcal{E}_{\max}$  is set to  $10^{-2}$ . With a number of poles  $\bar{n} = 5$ , the

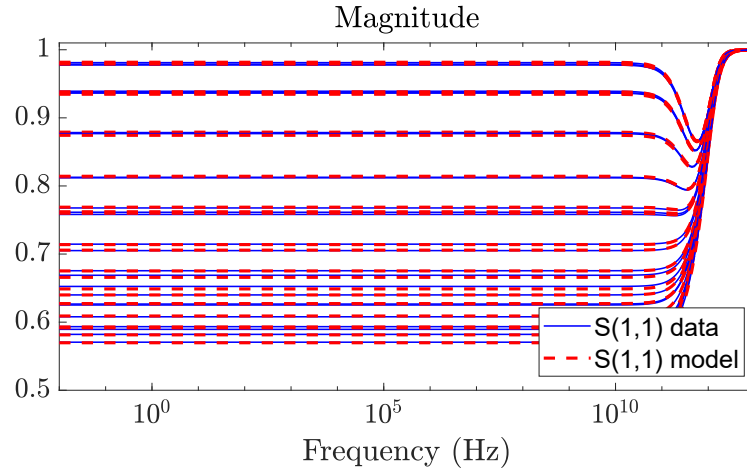


Fig. 6.7 Validation of the buffer model (red-dotted line) compared to raw data (blue solid line). From [119] © 2020 IEEE.

proposed approach extracts an accurate and uniformly stable model, whose worst-case absolute error on the validation samples is  $7.47 \times 10^{-3}$ , including 9 additional RBFs; the final model orders are therefore  $\bar{\ell}_N = \bar{\ell}_D = 19$ . The final model complexity, defined as  $\eta = \bar{\ell}_N + \bar{\ell}_D$ , is  $\eta = 38$ .

A comparison between model and data responses is provided in Fig 6.7.

The four panels in Figure 6.8 illustrate the adaptive steps over 4 iterations. The model-data error  $E^\nu(\boldsymbol{\vartheta}_m)$  evaluated on the current parametric validation samples is represented with a suitable color scale. The green-black dots depict the current set of RBF centers  $\mathcal{M}_\nu^c$  and the red-filled dot represent the new RBF center  $\tilde{\boldsymbol{\vartheta}}^\nu$  obtained as in (6.10). As expected, the adaptive algorithm locate new RBFs in correspondence to the largest error instance and obtains a uniformly accurate model in few iterations. The overall CPU time required to complete the optimization procedure is approximately 3 minutes.

## Numerical experiment (2) – a linearized low noise amplifier

In this second example, we consider the linearized 10-dimensional LNA model described in Section 5.3.2. We start with a set  $\mathcal{M}$  composed of  $M = 2000$  samples generated according to a Latin Hypercube sampling (see Section 5.3.2 and [101] for more details). To achieve the desired accuracy, the algorithm starts with  $m_0^c = 80$  RBF centers and  $m_0^t = 360$  training samples; the control

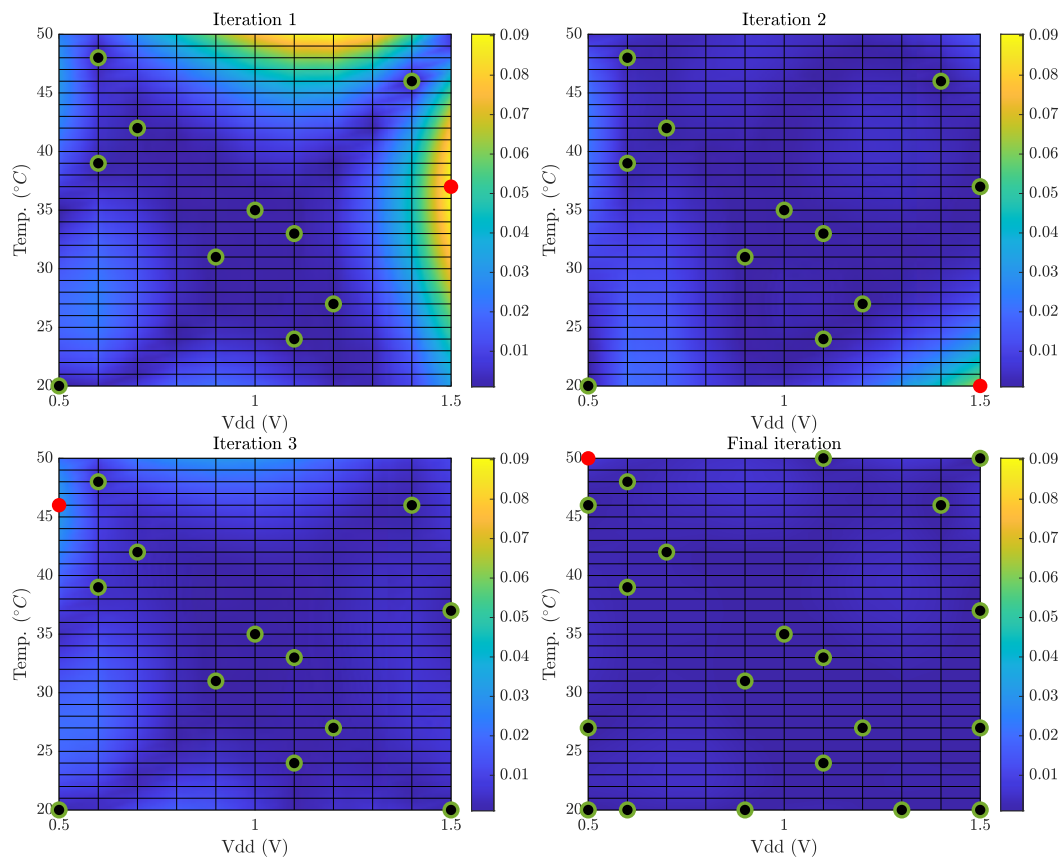


Fig. 6.8 Buffer model vs data error for some relevant algorithm iterations. Black dots depict the location of current RBF centers; red dots denote candidate new RBF centers.

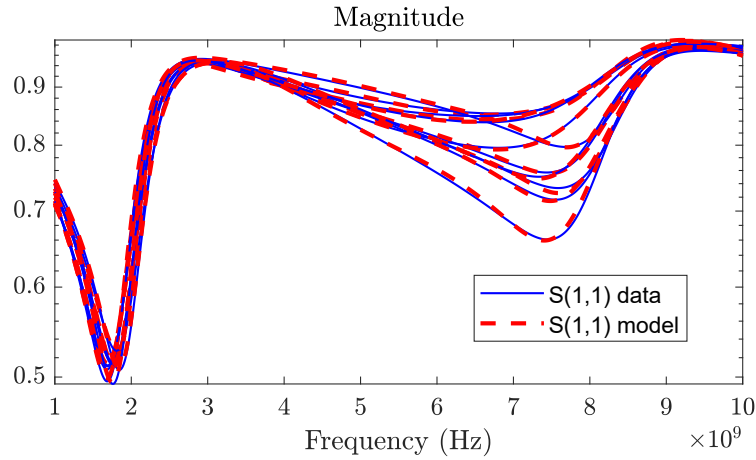


Fig. 6.9 Parametric model responses compared with a randomly selected subset of validation data.

parameters are set as  $\nu_{max} = 30$  and  $E_{max} = 10^{-2}$ . The model dynamic order is set to  $\bar{n} = 16$ ;

Differently from the previous bi-dimensional example, in high-dimensional space it is not possible to represent the error distribution as in Figure 6.8. For this example, we thus visualize the algorithm progress through histograms showing the distribution of the error over the validation samples.

As demonstrated in the four panels of Figure 6.9, the proposed algorithm placed 8 additional RBFs to obtain a uniformly stable model, whose accuracy is uniformly below the threshold  $E_{max}$  and equal to  $9.52 \times 10^{-3}$  in terms of absolute validation error. The final model complexity is  $\eta = 176$ .

Figure 6.9 compares the model responses with randomly chosen raw validation data. The adaptive augmentation procedure required approximately 35 minutes (using the optimized shape parameter selection presented in Section 6.2.3).

Together with the additional examples presented in [119], the above results demonstrate that the proposed concurrent optimization procedure enables the construction of uniformly accurate and stable macromodels, independently on the parameter space dimension.

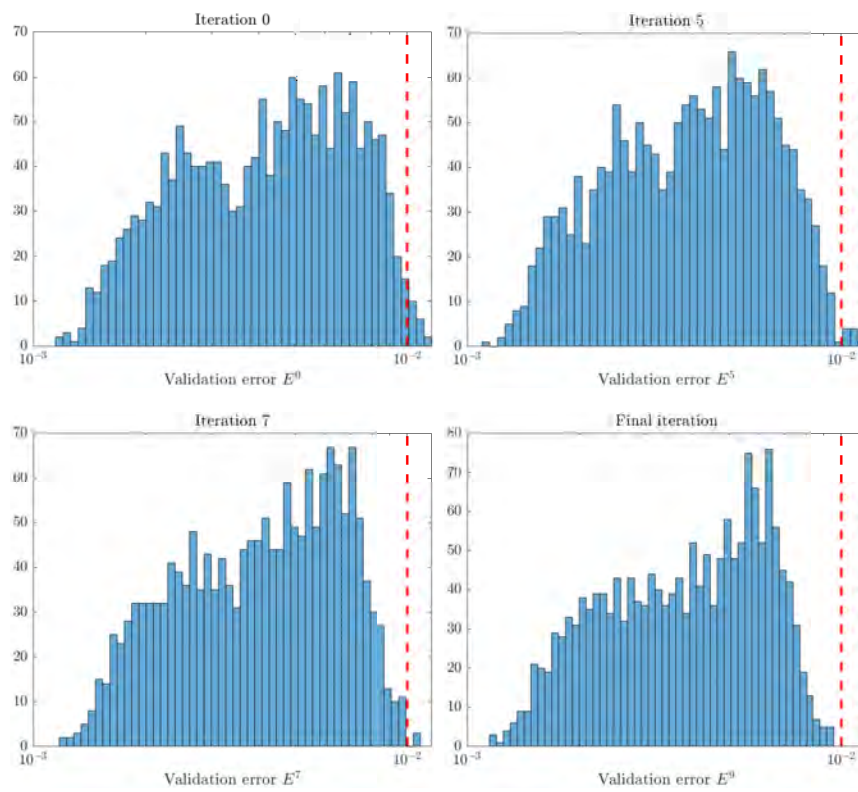


Fig. 6.10 Augmentation algorithm progresses through the adaptive iterations for the 10-parameters linearized LNA example.

## Discussion

We should point out that, although the adaptive procedure enables to add a reduced number of suitably positioned RBFs, the assumption of a common shared basis for numerator and denominator leads to unnecessarily complex models, especially when the parameter variability is complicated to track.

Another, perhaps more relevant, problem, stems from the need of computing in advance a large set of validation samples  $\mathcal{M}^v$  upon which we search the additional RBF centers. In fact, the generation of large amounts of parametric data may require extremely demanding repeated solver runs. This first approach is thus feasible whenever the extraction of parametric data is reasonably fast and when the (expected) model complexity is reasonably small.

### 6.3.2 Method 2: optimizing only the number of RBFs

Based on the above discussion, we are going to present a different strategy, originally presented in [103], developed in the interest of model compactness and computational runtime.

To this end, in this second method we do not seek for a sub-optimal RBFs distribution, but rather we assume a predefined distribution of the RBF centers, which is required only to cover as uniformly as possible the parameter space (Sobol sequences can therefore be good candidates).

**Remark 21.** *Although quite strong, this assumption is reasonable in our parameterized macromodeling setting; in fact, several numerical tests demonstrate that the improvement in model accuracy is not justified by the considerable computational effort required to optimize the RBFs' locations.*

The sequence of choice is then used for the expansion of both the numerator and the denominator that, in contrast with the previous method, are now considered independently, i.e.,  $\bar{\ell}_N$  may be different from  $\bar{\ell}_D$ . This additional degree of freedom will lead to a considerable reduction of the model complexity (see Section 6.3.2 for numerical results)

Let us therefore consider two sets of numerator and denominator candidate orders  $\mathcal{L}_N = \{\bar{\ell}_N^1, \dots, \bar{\ell}_N^r\}$  and  $\mathcal{L}_D = \{\bar{\ell}_D^1, \dots, \bar{\ell}_D^t\}$ , respectively, whose elements

are sorted in ascending order. In principle, the set of candidate orders should be chosen according to the parametric variability to approximate. However, this information is often unavailable. As a general rule, we noticed that the heuristic rule  $\bar{\ell}_N^1 = 4\rho$ ,  $\bar{\ell}_D^1 = 2\rho$  has proven to be successful in our test cases. In case additional information on the parametric variability are available, these can be used to facilitate the algorithm to find the most suitable orders.

Each element of the sets  $\mathcal{L}_N$ ,  $\mathcal{L}_D$  is associated with its corresponding centers, as

$$\begin{aligned} \mathcal{C}_N^r &= \{\boldsymbol{\vartheta}_m \in \mathcal{T} : m = 1, \dots, \bar{\ell}_N^r\} & r = 1, \dots, \bar{r} \\ \mathcal{C}_D^t &= \{\boldsymbol{\vartheta}_m \in \mathcal{T} : m = 1, \dots, \bar{\ell}_D^t\} & t = 1, \dots, \bar{t} \end{aligned} \quad (6.13)$$

for numerator and denominator, respectively. We recall that  $\mathcal{T}$  is the training set, composed of  $\bar{m}_t$  parametric responses used to fit the model. Following a procedure similar to the one exploited in the method 1, the elements of sets  $\mathcal{C}_N^r$ ,  $\mathcal{C}_D^t$  are selected as the nearest neighbors in  $\mathcal{T}$  of a truncated Sobol or Halton sequence. This choice enables a uniform coverage of the parameter space, which improves the numerical conditioning of the fitting problem and the accuracy of the resulting model. Then, once the number of numerator and denominator RBF is selected, their location on the parameter space is easily determined through (6.13).

We remark that the candidate orders in  $\mathcal{L}_{N,D}$  must fulfill two constraints

1. in order to ensure the applicability of the proposed algorithm for the selection of the shape parameter, we must guarantee that  $\bar{\ell}_{N,D}^1 > \rho + 2$ ;
2. in addition, we must guarantee that the model fitting problem is sufficiently over-determined by setting  $2(\bar{\ell}_N^{\bar{r}} + \bar{\ell}_D^{\bar{t}}) < \bar{m}_t$ .

Our objective is to find the most appropriate combination of model orders  $\bar{\ell}_N$ ,  $\bar{\ell}_D$  from sets  $\mathcal{L}_N$  and  $\mathcal{L}_D$  that maximizes the model-data accuracy. To this end, we use a *steepest-descent* strategy similar to [63], that proposes an order selection strategy in the context of parameterized Loewner macromodeling.

The number of numerator and denominator basis functions is determined exploiting a sub-optimal greedy approach, by updating at each iteration the numerator and denominator orders, denoted as  $\bar{\ell}_N^r$ ,  $\bar{\ell}_D^t$  and indexed by the pair



$(r, t)$ . In the following, we denote the model-data error evaluated at  $\bar{\ell}_N$  and  $\bar{\ell}_D$  as  $E(\bar{\ell}_N, \bar{\ell}_D, \varepsilon)$ .

**Remark 22.** *Each model extraction may require several PSK iterations to reach convergence. In order to improve the performances, we do not wait for convergence, and we perform only the first iteration. We consider the resulting model-data error as a proxy for the model accuracy attained at convergence; this is justified since the accuracy at convergence is higher with respect to the one at the first iteration.*

Further, optimizing the shape parameter through Algorithm 2, we introduce  $E^*(\bar{\ell}_N, \bar{\ell}_D) = \min_{\varepsilon} E(\bar{\ell}_N, \bar{\ell}_D, \varepsilon)$

At each greedy iteration, we evaluate the errors

$$E^*(\bar{\ell}_N^{r+\delta}, \bar{\ell}_D^t) \quad \text{and} \quad E^*(\bar{\ell}_N^r, \bar{\ell}_D^{t+\delta}), \quad (6.14)$$

where  $\delta = 1, \dots, \bar{\delta}$  is a predefined step size, and we update indices  $r$  and or  $t$  based on which combination provides the smallest error. The process is then repeated until a sub-optimal pair  $\{\bar{\ell}_N, \bar{\ell}_D\}$  is found.

In Figure 6.11 we illustrate the progresses of this order selection algorithm. Starting from  $\bar{\ell}_N^1 = 5$  and  $\bar{\ell}_D^1 = 5$ , the algorithm optimizes the number of RBFs following the (locally) the steepest descent path, identified with a red solid line. In this specific case, the lowest error instance is attained in correspondence of the maximum allowed orders, both at numerator and denominator; however, this is not always true (see, for instance, Figure 6.16).

### Numerical examples – comparing the two approaches

To assess the performance of this second strategy, we repeat the same test on the 10-parameters linearized LNA (see Section 5.3.2) used to test the first method, in order to provide a comparison between these two approaches. For a detailed description of the structure under modeling, we refer the Reader to Section 5.3.2.

We configured the shape parameter selection algorithm with the same settings used when testing the first concurrent optimization method and, follow-

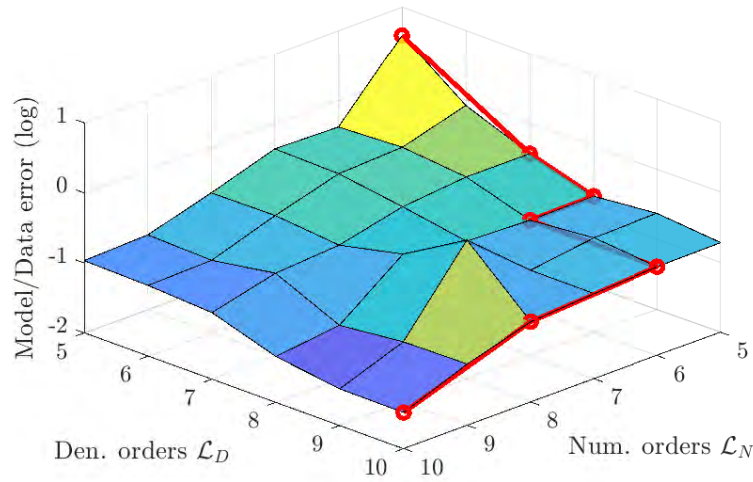


Fig. 6.11 Steps of the greedy RBF order selection algorithm. The surface represent the error  $E^*(\bar{\ell}_N, \bar{\ell}_D)$  after shape parameter optimization; the red thick line shows the path followed by the algorithm.

ing the proposed heuristic initialization rules, the steepest-descent algorithm searches on the sets  $\mathcal{L}_N = \{40, 41, \dots, 50\}$  and  $\mathcal{L}_D = \{20, 21, \dots, 30\}$ .

This second optimization method selected sub-optimal orders  $\bar{\ell}_N = 43$  and  $\bar{\ell}_D = 24$  which results in a uniformly stable macromodel whose accuracy is uniformly below 1%; Figure 6.12 compares the model responses for a randomly selected subset among the available  $\bar{m} = 2000$  raw data samples, confirming the accuracy. The overall CPU time to optimize the model orders is 13 minutes.

As anticipated, this strategy aims at optimizing the model structure, producing more compact models. Based on the model complexity  $\eta$  defined in Section 6.3.1, we recall the first concurrent optimization strategy provided a model with  $\eta = 176$ . In contrast, this second method resulted in  $\eta = 67$ . It is remarkable that, for the same model accuracy, the second approach provided a compression of approximately 62%.

In regard to the CPU time, the first approach requires 35 minutes, in contrast with this second that optimized the orders in 13 minutes, resulting in a speed-up of approximately 3 times.

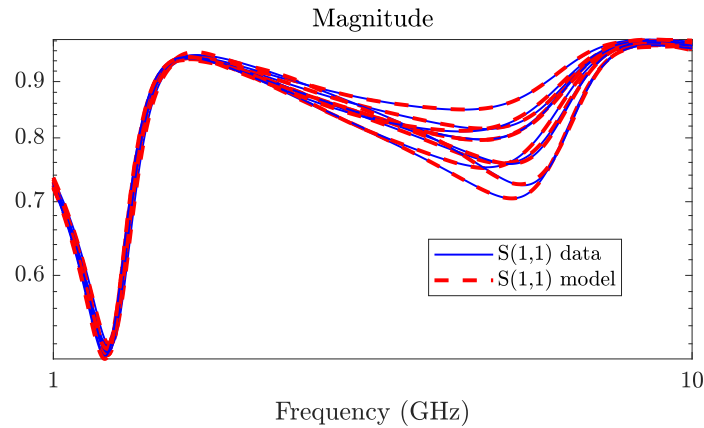


Fig. 6.12 Comparing the model responses over a random subset of validation samples for the LNA test-case. From [103] © 2021 IEEE

## 6.4 A fully-automated macromodeling flow

After the introduction of efficient algorithmic strategies for the automated optimization of the RBFs' hyper-parameters, we are very close to set up a fully automated high-dimensional macromodeling strategy. The only missing step is the optimization of the number of poles  $\bar{n}$ , which we address below.

### 6.4.1 Optimizing the number of model poles

The problem of selecting a suitable dynamic order  $\bar{n}$  has been already solved in the context of non-parameterized macromodeling. Following similar techniques, we present a heuristic parameterized extension that, despite its simplicity, provides good results in acceptably short runtimes.

The estimation of a proper dynamic order  $\bar{n}$  is performed iteratively in a pre-processing step. First, we define and collect a set of candidate orders  $\{\bar{n}_1, \dots, \bar{n}_q\}$ , sorted in ascending order. Then, among all the available data samples, we select a significant subset of frequency responses associated to some parameter values  $\boldsymbol{\vartheta}_z$

$$\mathcal{P} = \{\boldsymbol{\vartheta}_z \in \mathcal{T} : z = 1, \dots, \bar{z}\}. \quad (6.15)$$

Our strategy is to collect in  $\mathcal{P}$  the parameter locations corresponding to the nearest neighbors in  $\mathcal{T}$  to the vertices and centroid of the parameter space. We remark that this choice is not restrictive.

By iteratively increasing the order  $\bar{n} = \bar{n}_1, \bar{n}_2, \dots$ , we independently apply the VF algorithm to all the  $\bar{z}$  frequency responses collected corresponding to the parameter instances in set  $\mathcal{P}$ . It is expected that, as the iterations  $I$  proceed, the worst case model-data error  $\mathcal{E}_I$ , evaluated among all the  $\bar{z}$  elements, reduces. The iterations stop at  $I = I^*$  when  $\mathcal{E}_{I^*} < \mathcal{E}_{th}$ , with  $\mathcal{E}_{th}$  a predefined threshold. The corresponding order  $\bar{n} = \bar{n}_{I^*}$  is selected, and the common basis poles  $\{q_n\}$  are selected as the poles of the vector fitting model associated to the nearest neighbor to the centroid of the parameter space. In Figure 6.15 we illustrate the performances of this algorithm on a relevant example.

## 6.4.2 An algorithm for automated macromodeling

Now we have all the building blocks to develop an automated procedure to construct high-dimensional (guaranteed stable) parameterized macromodels.

**Remark 23.** *In this work, we assume that the parameterized raw data are available, and no additional on demand solver calls are required.*

The proposed strategy is illustrated in the block diagram of Figure 6.13. Starting from the parametric raw data  $\check{\mathbf{H}}_{k,m}$ , we first determine the dynamic order as described in Section 6.4.1. As  $\bar{n}$  is available, we estimate the RBFs' locations and their number with the strategies elaborated in Section 6.3. Then, we compute a sub-optimal shape parameter value through Algorithm 2, that is finally used to construct a parameterized uniformly stable macromodel  $\mathbf{H}(s, \boldsymbol{\vartheta})$ .

On the left, we provide a list of the algorithm's control parameters for each modeling step. We remark that, although some control parameters are inherently case-dependent i.e.,  $\mathcal{L}_N, \mathcal{L}_D$ , for all the others there are trade-off configurations that turned out to be suitable for all the considered test-cases. Expert users can thus finely tune these additional parameters to optimize the modeling performances; conversely, unexperienced users may still successfully use the proposed procedure with standard settings, at the cost of a slight performance degradation.

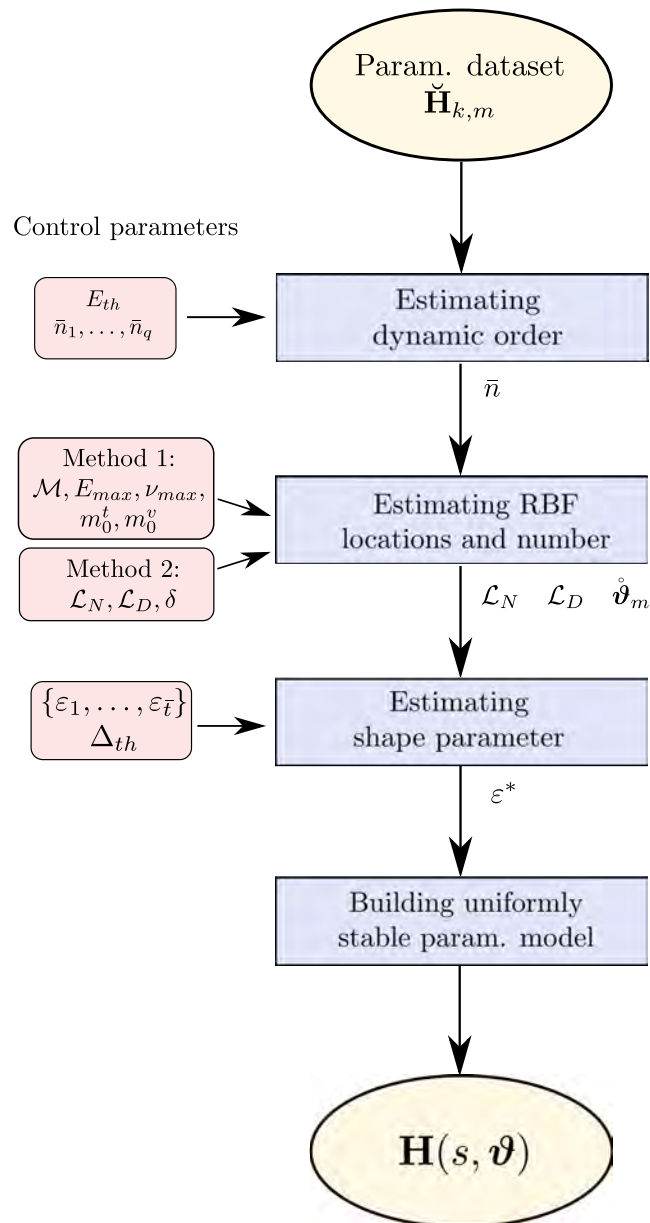


Fig. 6.13 High-level block diagram of the proposed automated parameterized macromodeling flow.

### 6.4.3 Numerical experiments:

We are going to test the proposed automated macromodeling scheme on numerical examples. All the results have been obtained on a standard 16 GB commercial laptop, running an 2.30 GHz i7 Intel processor.

#### A multi-board interconnect

To begin, we consider a reduced complexity bi-dimensional ( $\rho = 2$ ) test-case, in order to better illustrate our proposed scheme. The device under modeling is the multi-board interconnection introduced in Chapter 4. We refer the reader to Section 4.5.2 and [3] for detailed information.

The proposed automated macromodeling strategy has been executed using the following control parameters.

1. *Dynamic order selection:*

The set of candidate orders is  $\{10, 11, \dots, 29, 30\}$ , and the stopping threshold has been set to  $\mathcal{E}_{th} = 10^{-3}$ .

2. *Parameter order selection:*

In all the proposed examples, the RBFs' location and number will be optimized using the second approach (optimization of only the number of RBFs). Here, we set  $\mathcal{L}_N = \mathcal{L}_D = \{5, \dots, 10\}$  and use a step size  $\bar{\delta} = 2$ .

3. *Shape parameter selection:*

The shape parameter has been selected among 10 logarithmically spaced candidates ranging from  $10^{-3}$  to 10, and using a stopping threshold  $\Delta_{th} = 4 \times 10^{-2}$ .

Based on these settings, the proposed scheme returned the sub-optimal hyper-parameter listed in Table 6.1. The hyper-parameters optimization and the model synthesis procedure cumulatively required only 41 seconds. The worst-case model-data error with respect to validation samples is  $7 \times 10^{-3}$ . The accuracy is confirmed by Figure 6.14, that compares the modeled transmission S-parameters corresponding to a randomly selected subset of validation data.

Table 6.1 Hyper-parameters selected by proposed algorithm for the multi-board interconnect.

Number of poles	$\bar{n}$	24
Numerator parameter order	$\bar{\ell}_N$	10
Denominator parameter order	$\bar{\ell}_D$	8
Numerator shape parameter	$\varepsilon_N^*$	0.4642
Denominator shape parameter	$\varepsilon_D^*$	0.4642

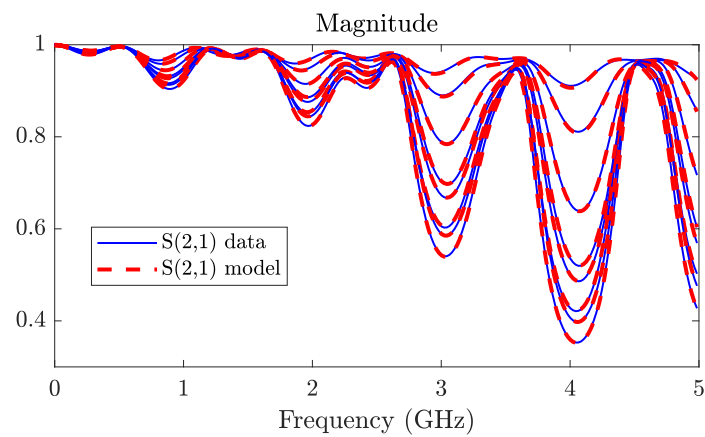


Fig. 6.14 Model responses compared with validation data for the multi-board interconnect example. Blue lines: randomly-selected validation data. Red dashed lines: corresponding model responses. From [103] © 2021 IEEE

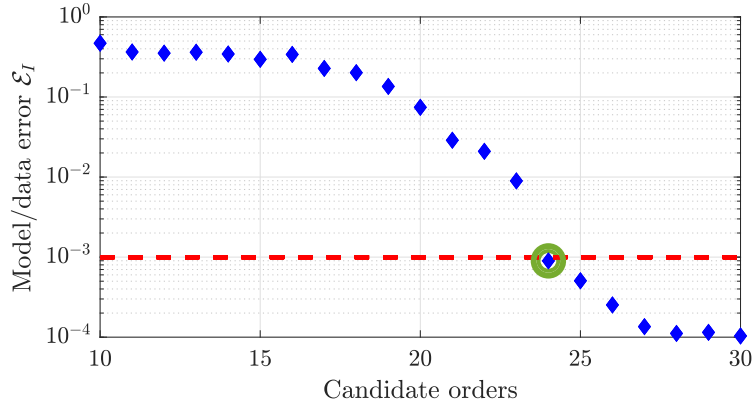


Fig. 6.15 Multiboard interconnect: iterations of the dynamic order selection algorithm. Blue markers: worst-case model data error (among the selected parameter values) for subsequent candidate number of poles; red-dashed line: stopping threshold; green circle: selected dynamic order. From [103] © 2021 IEEE

We now assess the performances of all the hyper-parameters optimization steps, by comparing the obtained estimates with the results given by an exhaustive grid search.

1. *Dynamic order selection:*

As demonstrated in Figure 6.15, starting from  $\bar{n}_1 = 10$ , the algorithm iteratively increases the number of poles, until the stopping condition  $E_{I^*} < E_{th}$  is reached at iteration  $I^* = 15$  (depicted with a green circle), that corresponds to  $\bar{n} = 24$ . It is worth noting that the error saturates at  $10^{-4}$  for large  $\bar{n}$ , highlighting the presence of non-causal or noisy components in the data. This sets a limit on the attainable accuracy, which must be taken into consideration when setting  $\mathcal{E}_{th}$ .

2. *Parameter order selection:*

In Figure 6.11 we represent the iterative steps of the order selection algorithm of Section 6.3.2. Starting from  $\bar{\ell}_N^1 = \bar{\ell}_D^1 = 5$ , it iteratively follows the locally optimal path until it stops at  $\bar{\ell}_N^5 = 10$ ,  $\bar{\ell}_D^5 = 8$  after 5 iterations.

3. *Shape parameter selection:*

Similarly to Figure 6.6, in Fig. 6.17 we compare the shape parameter estimates  $\{\varepsilon_N, \varepsilon_D\}$  obtained through the proposed Algorithm 2 (green cross) with the reference solution obtained with grid search (red circle).



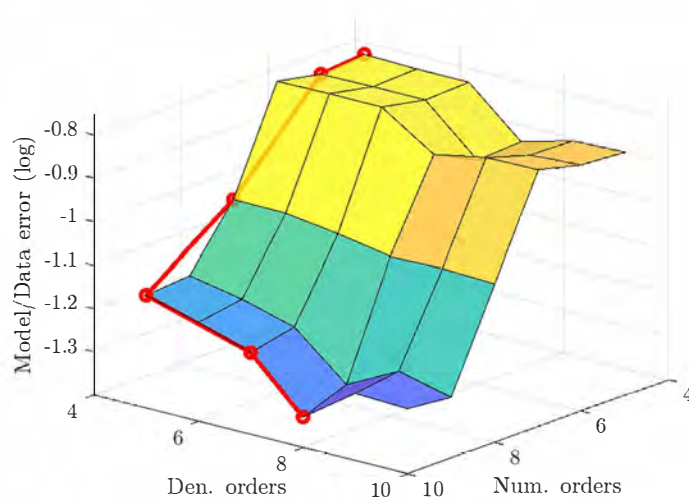


Fig. 6.16 The same as in Figure 6.11, but for the multi-board interconnect example. From [103] © 2021 IEEE

The model-data accuracy, evaluated for different  $\{\varepsilon_N, \varepsilon_D\}$  combinations is represented with a suitable color-scale. The estimates computed with our heuristic approach (green cross) coincide with the reference grid search (red circle) and are equal to  $\varepsilon_N^* = \varepsilon_D^* = 0.4642$ .

Let us now compare the computational performances. In particular, the proposed order optimization strategy is twice as fast with respect to a standard grid search. However, the most striking results pertain to the sub-optimal shape parameters  $\varepsilon_N^*$  and  $\varepsilon_D^*$ , whose optimization through Algorithm 2 resulted to be approximately 2000 times faster with respect to a grid search over a  $10 \times 10$  grid.

### A transmission line network

As a second and last example, we model the distributed transmission line network illustrated in Figure 6.18 composed of four cascaded lossy microstrip segments (with length  $l_i$ ) and three internal loaded stubs (with length  $s_i$ ). The purely resistive loads have values  $R_i$ . The copper microstrip has fixed thickness  $t = 30 \mu\text{m}$  and parameterized width  $w$ , and the dielectric substrate on which it is placed has a variable height  $h$ , a fixed relative dielectric constant  $\varepsilon_r = 4.1$  and

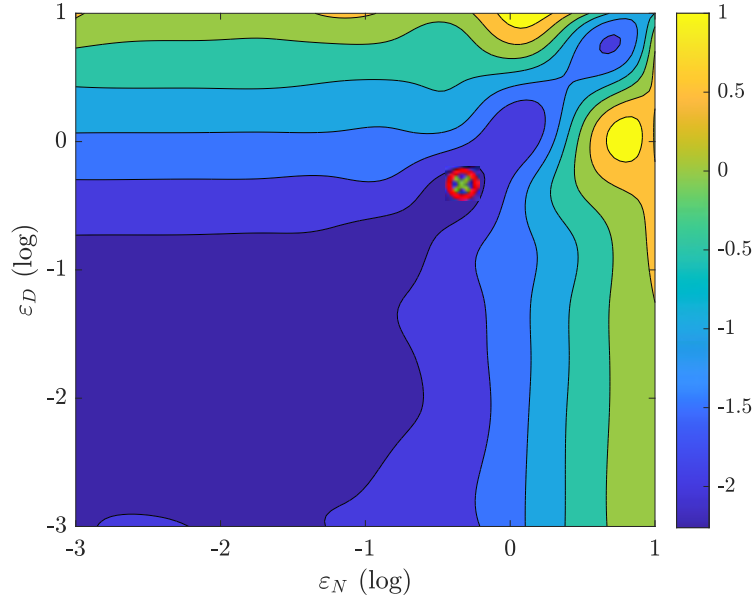


Fig. 6.17 Shape parameter selection algorithm. The contour plot represents the model-data error as a function of numerator and denominator shape parameter values  $\varepsilon_N, \varepsilon_D$ . The green cross pinpoints the sub-optimal combination  $\varepsilon_N^*, \varepsilon_D^*$  selected by the proposed algorithm. From [103] © 2021 IEEE

loss tangent  $\tan \delta = 0.02$ . We remark that, although it is structurally similar to the one presented in Section 4.5.3, the parametric raw data are obtained using different line models and a different solver.

The range of variation  $\Delta\vartheta^i$  of the above 10 independent parameters  $\vartheta^i$  are reported in Table 6.2, together with their nominal values  $\bar{\vartheta}^i$ . We highlight that, in order to ensure the total line length to be always equal to 15 mm, the line lengths  $l_i$  are subject to the constraint  $l_1 + l_2 + l_3 + l_4 = 15$  mm.

The parameterized dataset has been obtained by repeated HSPICE [112] runs. In particular, it is composed of  $\bar{m} = 300$  S-parameter frequency responses, each containing  $\bar{k} = 401$  linearly spaced samples in the band  $\Omega = [0, 20]$  GHz. The hyper-parameter estimation algorithms has been set as in the previous example, with the exception for  $\mathcal{L}_N$  and  $\mathcal{L}_D$  that now span the intervals  $[40, 41, \dots, 50]$  and  $[20, 21, \dots, 30]$ , respectively.

The proposed algorithm automatically determined a model structure with  $\bar{n} = 27$  poles, numerator and denominator orders  $\bar{\ell}_N = 40$  and  $\bar{\ell}_D = 27$ , respectively, with associated shape parameter values  $\varepsilon_N^* = \varepsilon_D^* = 0.0599$ . Altogether, the optimization of the hyper-parameters required approximately 14 minutes,

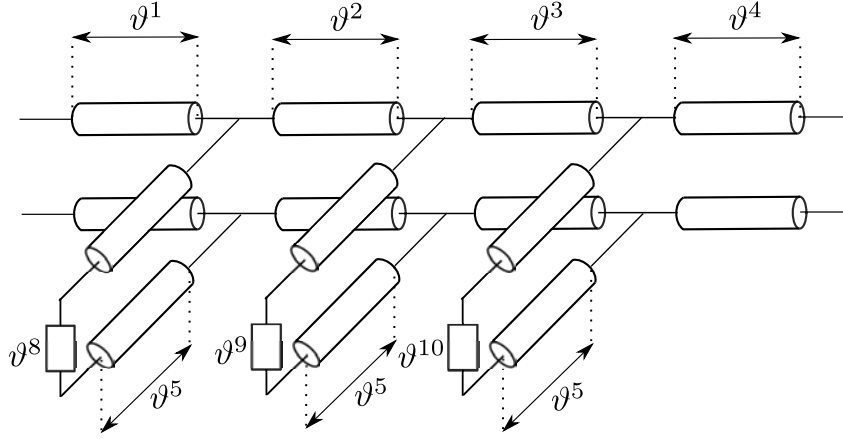


Fig. 6.18 Illustration of the 10-dimensional transmission line network used to test the proposed automated macromodeling framework.

Table 6.2 Transmission line network parameters.  $l_i, s_i$ : length of transmission line and stub segments;  $w$ : conductor width;  $h$ : substrate height;  $R_i$ : stub termination resistances.

#	Parameter $\vartheta^i$	$\bar{\vartheta}^i$	$\Delta\vartheta^i$
1	$l_1$ (mm)	10	5 %
2	$l_2$ (mm)	10	5 %
3	$l_3$ (mm)	10	5 %
4	$l_4$ (mm)	10	5 %
5	$s_{1,2,3}$ (mm)	3.75	2 %
6	$w$ ( $\mu\text{m}$ )	150	10 %
7	$h$ ( $\mu\text{m}$ )	200	10 %
8	$R_1$ ( $\Omega$ )	50	10 %
9	$R_2$ ( $\Omega$ )	50	10 %
10	$R_3$ ( $\Omega$ )	50	10 %

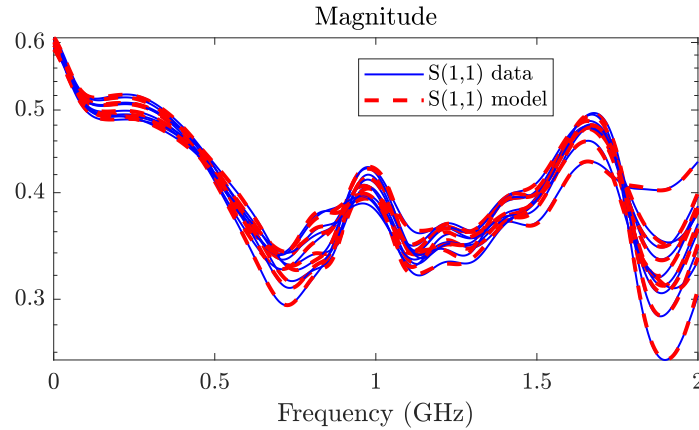


Fig. 6.19 Transmission line network: validation of the parameterized model responses (red dashed lines) through corresponding reference data (blue solid lines). From [103] © 2021 IEEE

while it took only 30 seconds for the construction of the uniformly stable model, whose accuracy with respect to raw validation data is  $11.2 \times 10^{-3}$  and its complexity is  $\eta = 69$ . In Figure 6.19, we demonstrate the accuracy of the model, with respect to a subset of validation data.

As a comparison, if the order selection had been performed following the first “concurrent optimization” method, the extraction of a model with comparable accuracy would have required 1.5 hours (6 times slower), whose complexity  $\eta$  would have been 112 (39% larger).

## 6.5 Passivity for high-dimensional models

The above examples demonstrate the capability of the presented macromodeling flow to build compact yet accurate high-dimensional models, that are guaranteed to be uniformly stable throughout the parameter space. Therefore, high-dimensional models of (linearized) active circuit blocks synthesized with this approach can be reliably used as accurate surrogates in parametric time-domain simulations, without the risk of running into unstable behaviors.

As we thoroughly illustrated in Chapter 3, behavioral modeling of passive components requires stronger conditions: models of passive components *must* reflect this property in order to ensure stable and reliable simulations. As discussed in Chapter 4, model passivity can be enforced through post processing

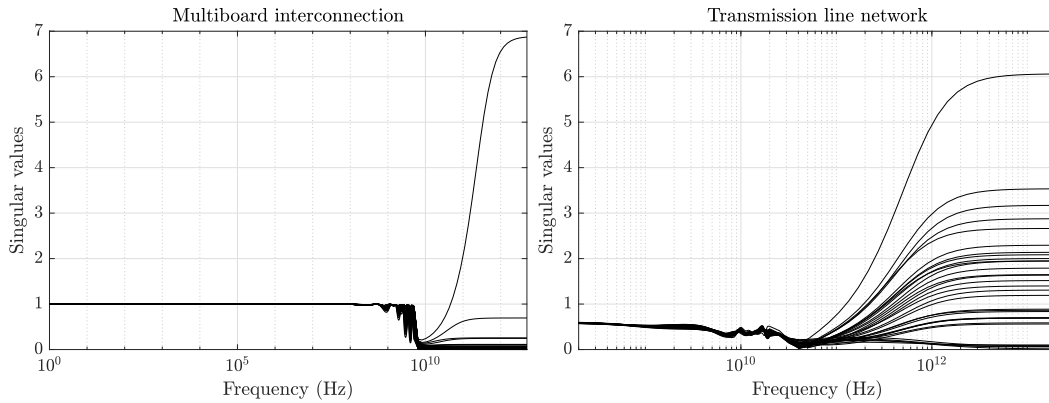


Fig. 6.20 Parameterized singular values associated with the original non-passive models. Left panel: multi-board interconnection. Right panel: transmission line network.

even in a high-dimensional setting. In fact, the passivity enforcement algorithm through iterative local perturbation discussed in Section 4.4 is applicable without modification. As a prerequisite, this algorithm requires a precise identification of the passivity violations in the frequency-parameter space. This characterization becomes extremely critical when the parameters exceed few units, in which case no deterministic algorithms currently exist that are able to identify all violations. In particular, all passivity characterizations presented in Chapter 4 are not scalable to high dimensions.

A promising yet preliminary approach to multi-variate passivity characterization with good potential for improved scalability to high-dimensions has been recently presented in [133]. This approach performs a simple tree-based adaptive sampling in the frequency-parameter space. For this reason we do not include additional details here referring the Reader to [133]. This algorithm was incorporated in the proposed automated macromodeling flow; some examples and results are discussed next.

In the following, we present results for the above multi-board interconnect and transmission line networks examples. Figure 6.20 highlight the presence of relevant out-of-band violations. Applying the proposed iterative passivity enforcement procedure we obtained models of comparable accuracy (Fig. 6.21), for which the passivity check procedure [133] does not find residual violations, as illustrated in Figure 6.22.

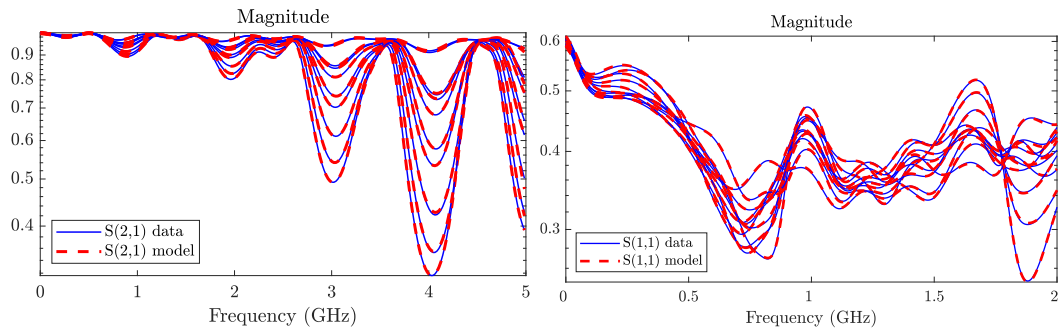


Fig. 6.21 Comparing the accuracy of candidate passive models with respect to a randomly selected subset of validation data.

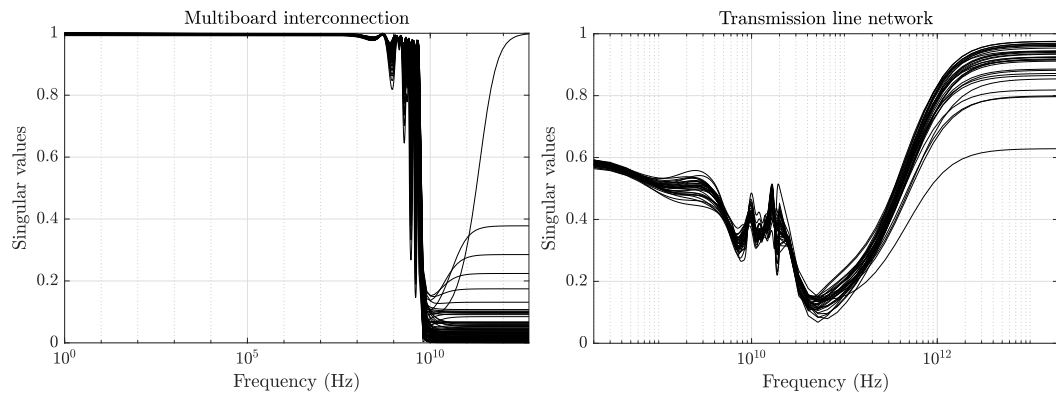


Fig. 6.22 Parameterized singular values associated with the original non-passive models. Left panel: multi-board interconnection. Right panel: transmission line network.

Despite the results of Figure 6.22, however, we can not claim that the models are passive since the verification is based on a finite number of samples computed in the frequency-parameter space.

In conclusion, based on the theoretical and numerical results presented in this work we see that the only major problem that still remains unsolved is passivity verification in high-dimensional parameterized setting. This problem will be the subject of future investigations.

# Chapter 7

## Conclusions

This dissertation addressed the topic of parameterized data-driven macromodeling. In particular, a set of algorithms have been developed for the extraction of reduced-order models of complex linear and time-invariant structures, whose behavior depends on possibly several external parameters. Data-driven model order reduction has been considered, so that the models are extracted from a limited set of frequency-domain sampled responses, available from a first-principle description (e.g., a full-wave electromagnetic solver). Building upon the well-established *Parameterized Sanathanan Koerner* (PSK) model structure and identification procedure, this thesis provides a comprehensive set of techniques that enable automated extraction of guarantee stable and passive parameterized behavioral models. The specific contributions are itemized below.

**Multivariate extension** The construction of parameterized macromodels from sampled responses is not a difficult task in the well-established PSK framework. The real challenge is to provide an explicit certification of stability and passivity, which are fundamental properties that the models must have in order to be of any use in practical design flows. These aspects are well consolidated for non-parameterized models, but prior to this work only partial solutions were available in a general parameterized (multivariate) case. In particular, passivity verification was only limited to a single external parameter.

This problem has been addressed in Chapter 4, where we proposed a theoretical framework and the associated algorithms for checking and enforcing



both stability and passivity of general multivariate parameterized models. The main tool is a formulation of a parameter-dependent Hamiltonian matrix, whose properties are exploited for the precise identification of stability and passivity violation regions in the frequency-parameter space, in view of their elimination by iterative perturbation. The main limitation of this approach is related to the inevitable *curse of dimensionality*, which prevents scalability to high dimensions and grants applicability only up to three independent external parameters.

**High-dimensional parameterization** Pushing further with the number of parameters would bring an enhanced flexibility in parametric simulations, in which a single compact macromodel would be representative of a much broader spectrum of design configurations and working conditions. The standard PSK model structure demands for identification procedures that may become exponentially more complex as the number of parameter increases. We fill this technical gap in Chapter 5, where we introduce a novel unstructured high-dimensional model functional form based on *Radial Basis Functions* (RBFs), specifically tailored to embed a much larger number of parameters in the model. In addition, we introduce novel uniform stability constraints, whose formulation does not depend on the dimension of the parameter space, but only on the complexity of the model, here defined in terms of the number of numerical coefficients that are determined during identification. Despite the increased degree of conservativeness of such constraints, excellent results in terms of accuracy and scalability have been demonstrated on benchmark examples with up to ten independent parameters.

**Automation** The proposed high-dimensional RBF parameterization requires the selection of a number of hyper-parameters on which the model structure is built. These include: dynamic order (number of poles), dimension of the approximation spaces for the numerator and denominator of the model transfer function, centers and shape parameters defining the individual radial basis functions used in the approximation. Trial and error approaches, as well as naive grid search techniques, are extremely time consuming and not compatible with an efficient identification process. In Chapter 6, we have presented a comprehensive set of algorithms for the automated estimation of sub-optimal hyper-parameter configurations. This estimation does not require a large

number of model extractions hence is very fast and effective, leading to accurate parameterized models basically without any user interaction.

**Impact** This dissertation provided advancements under both theoretical and practical standpoints. In particular, the availability of an automated parameterized macromodeling tool can be of great advantage both for industries, who could exploit the proposed framework to generate models for improving reliability and efficiency of the design workflows, by replacing costly full-wave or first-principle circuit simulation with robust equivalent reduced-order surrogate models. Also, Electronic Design Automation (EDA) software companies may directly benefit of the proposed framework to include parameterized reduced order models to improve the performances of their commercial tools, for instance in all those situations where the performance of some structure needs to be optimized by tuning geometry or material parameters, or when expensive sensitivity or monte-carlo loops are required. Availability of parameterized surrogate models may dramatically reduce runtime while preserving accuracy and reliability of the results. The framework discussed in this dissertation is ready for exploitation in such application scenarios.

## Open problems

Despite the good results documented in this dissertation, several open problems remain. The most relevant are itemized below.

**High-dimensional passivity enforcement** The automated macromodeling flow presented in Chapter 6 has demonstrated to be very effective for the synthesis of accurate and guaranteed stable high-dimensional models. However, the lack of a uniform passivity certificate restricts its applicability to active (linearized) devices, ruling out the extremely broad and important category of passive structures.

In fact, although in Chapter 4 we presented a passivity verification and enforcement scheme that extends the results of prior research works to higher-dimensional multivariate macromodels, its applicability is still limited to small

---

sized parameter spaces, due to the adopted sub-gridding technique that inherently suffers from the curse of dimensionality.

Conversely, the results presented in Chapter 6 are extremely promising. Indeed, the tree-based passivity verification algorithm presented in [133] has natively a good potential of improved scalability in high-dimensions. When employed in conjunction with the proposed iterative passivity enforcement scheme, it provided encouraging results. However, these are still preliminary and further investigations are required to assess the performances and the reliability of this scheme.

Alternatively, instead of guaranteeing passivity through a post-processing enforcement step, a recent research work [86] proposed a novel PSK implementation that guarantees uniform passivity by construction. Unfortunately, despite the remarkable results, this approach relies on model forms built upon multi-variate polynomial parameter basis functions, that poorly fit our high-dimensional framework. Future researches will be devoted to assess the compatibility of this innovative strategy with the proposed high-dimensional structure.

We can thus conclude that, although this work extended the previous results to more than one parameter, the problem of guaranteeing passivity of high-dimensional parameterized macromodels (tens of independent parameters) must be still regarded as an open research challenge.

**Adaptive macromodeling** Throughout this work, we have always assumed to be given with a prescribed set of parametric data, that have been used to train the parameterized macromodels. However, as pointed out in Chapter 1, the data-generation step is extremely critical, as it demands for repeated runs of, e.g., full-wave field solvers, that are highly demanding in terms of CPU resources. Therefore, gathering a minimal set of data samples necessary to build accurate models would be very helpful.

In case of very few parameters, this problem has been tackled in [46, 134] by employing an adaptive exploration of the parameter space. These works propose to interact with the EM solver, by iteratively collecting new data, based on some heuristics to determine the most relevant samples which best characterize the input/output responses of interest. However, these schemes rely on structured

subdivisions of the parameter space and can not be successfully employed in high-dimensional spaces due to an exponentially increasing computational complexity.

In the considered high-dimensional setting, the problem of gathering a minimal set of training data is still partially unsolved. Some notable approaches based on unstructured adaptive schemes such, for instance, Gaussian Processes [102] or some form of Bayesian Optimization, are known to have good potential to scale favorably in high-dimensions. Exploitation of these methods within the parameterized macromodeling framework is therefore regarded as a future research topic with high potential.

# References

- [1] IdEM R2012, IdemWorks s.r.l. [online] <http://www.idemworks.com>.
- [2] Sigrity BroadBand Spice, Cadence [online]. [https://www.cadence.com/ko\\_KR/home/tools/ic-package-design-and-analysis/si-pi-analysis-point-tools/sigrity-broadband-spice.html](https://www.cadence.com/ko_KR/home/tools/ic-package-design-and-analysis/si-pi-analysis-point-tools/sigrity-broadband-spice.html). Accessed: 2022-06-07.
- [3] JB Preibisch, T Reuschel, K Scharff, J Balachandran, B Sen, and C Schuster. Exploring efficient variability-aware analysis method for high-speed digital link design using pce. *DesignCon, Jan*, 2017.
- [4] Stefano Grivet-Talocia and Riccardo Trincherò. Behavioral, parameterized, and broadband modeling of wired interconnects with internal discontinuities. *IEEE Transactions on Electromagnetic Compatibility*, 60(1):77–85, 2018.
- [5] Chiara Ramella, Alessandro Zanco, Marco De Stefano, Tommaso Bradde, Marco Pirola, and Stefano Grivet-Talocia. Efficient em-based variability analysis of passive microwave structures through parameterized reduced-order behavioral models. In *2022 17th European Microwave Integrated Circuits Conference (EuMIC)*, 2022. accepted.
- [6] Keysight ADS, Keysight [online]. <https://www.keysight.com/>. Accessed: 2022-06-07.
- [7] Francesco Ferranti, Luc Knockaert, and Tom Dhaene. Passivity-preserving parametric macromodeling by means of scaled and shifted state-space systems. *IEEE Transactions on Microwave Theory and Techniques*, 59(10):2394–2403, 2011.
- [8] Francesco Ferranti, Luc Knockaert, Tom Dhaene, and Giulio Antonini. Passivity-preserving parametric macromodeling for highly dynamic tabulated data based on lur’e equations. *IEEE Transactions on Microwave Theory and Techniques*, 58(12):3688–3696, 2010.
- [9] E. R. Samuel, L. Knockaert, F. Ferranti, and T. Dhaene. Guaranteed passive parameterized macromodeling by using Sylvester state-space realizations. *Microwave Theory and Techniques, IEEE Transactions on*, 61(4):1444–1454, April 2013.

- 
- [10] D. Deschrijver, B. Haegeman, and T. Dhaene. Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses. *Advanced Packaging, IEEE Transactions on*, 30(2):216–225, may 2007.
- [11] P. Triverio, S. Grivet-Talocia, and M. S. Nakhla. A parameterized macromodeling strategy with uniform stability test. *IEEE Trans. Advanced Packaging*, 32(1):205–215, Feb 2009.
- [12] Michael Mongillo et al. Choosing basis functions and shape parameters for radial basis function methods. *SIAM undergraduate research online*, 4(190-209):2–6, 2011.
- [13] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.
- [14] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, pages 507–523. Springer, 2011.
- [15] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, et al. Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *arXiv preprint arXiv:2107.05847*, 2021.
- [16] Sanda Lefteriu and Stefano Grivet-Talocia. Broadband modeling of passive components via augmented equivalent circuit models. In *2018 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pages 1–2, 2018.
- [17] B. Gustavsen and A. Semlyen. Enforcing passivity for admittance matrices approximated by rational functions. *Power Engineering Review, IEEE*, 21(2):54, feb. 2001.
- [18] S. Grivet-Talocia. Passivity enforcement via perturbation of Hamiltonian matrices. *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, 51(9):1755–1769, September 2004.
- [19] Stefano Grivet-Talocia and Bjorn Gustavsen. *Passive macromodeling: Theory and applications*. John Wiley & Sons, 2015.
- [20] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: Passive reduced-order interconnect macromodeling algorithm. *IEEE transactions on computer-aided design of integrated circuits and systems*, 17(8):645–654, 1998.

- 
- [21] Serkan Gugercin and Athanasios C Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004.
- [22] J. R. Phillips and L. M. Silveira. Poor man’s TBR: a simple model reduction scheme. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(1):43–55, 2005.
- [23] A. C. Antoulas. *Approximation of large-scale dynamical systems*. Society for Industrial and Applied Mathematics, 2005.
- [24] P. Benner, M. Hinze, and E. J. W. ter Maten. *Model reduction for circuit simulation*, volume 74. Springer, 2010.
- [25] W. H. A. Schilders, H. A. Van Der Vorst, and J. Rommes. *Model order reduction: theory, research aspects and applications*. Springer Verlag, 2008.
- [26] S. Grivet-Talocia. Package macromodeling via time-domain vector fitting. *IEEE Microwave and Wireless Components Letters*, 13(11):472–474, November 2003.
- [27] S. Grivet-Talocia. The time-domain vector fitting algorithm for linear macromodeling. *Int. Journal of Electronics and Communications (AEU)*, 58:293–295, 2004.
- [28] Tommaso Bradde, Samuel Chevalier, Marco De Stefano, Stefano Grivet-Talocia, and Luca Daniel. Handling initial conditions in vector fitting for real time modeling of power system dynamics. *Energies*, 14(9), 2021.
- [29] Jian Li, Yufa Teng, Qingling Zhang, Jinghao Li, and Liang Qiao. Eliminating impulse for descriptor system by derivative output feedback. *Journal of applied mathematics*, 2014, 2014.
- [30] T. Kailath. *Linear systems*. Prentice-Hall Englewood Cliffs, NJ, 1980.
- [31] E. C. Levy. Complex curve fitting. *IRE Trans. on Automatic Control*, 4:37–44, 1959.
- [32] C. Sanathanan and J. Koerner. Transfer function synthesis as a ratio of two complex polynomials. *Automatic Control, IEEE Transactions on*, 8(1):56–58, jan 1963.
- [33] Jeffrey M Hokanson and Caleb C Magruder. Least squares rational approximation. *arXiv preprint arXiv:1811.12590*, 2018.
- [34] B. Gustavsen. Relaxed vector fitting algorithm for rational approximation of frequency domain responses. In *Signal Propagation on Interconnects, 2006. IEEE Workshop on*, pages 97–100, may 2006.

- 
- [35] Sanda Lefteriu and Athanasios C Antoulas. On the convergence of the vector-fitting algorithm. *IEEE Transactions on Microwave Theory and Techniques*, 61(4):1435–1443, 2013.
- [36] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *Power Delivery, IEEE Transactions on*, 14(3):1052–1061, jul 1999.
- [37] B. Gustavsen. Frequency-dependent transmission line modeling utilizing transposed conditions. *Power Delivery, IEEE Transactions on*, 17(3):834–839, jul 2002.
- [38] S. Lefteriu and A. C. Antoulas. On the convergence of the vector-fitting algorithm. *Microwave Theory and Techniques, IEEE Transactions on*, 61(4):1435–1443, 2013.
- [39] Branko Grünbaum, Victor Klee, Micha A Perles, and Geoffrey Colin Shephard. *Convex polytopes*, volume 16. Springer, 1967.
- [40] S. Grivet-Talocia and M. Bandinu. Improving the convergence of vector fitting for equivalent circuit extraction from noisy frequency responses. *IEEE Transactions on Electromagnetic Compatibility*, 48(1):104–120, 2006.
- [41] B. Gustavsen. Improving the pole relocating properties of vector fitting. *Power Delivery, IEEE Transactions on*, 21(3):1587–1592, july 2006.
- [42] A. C. Antoulas and B. D. O. Anderson. On the scalar rational interpolation problem. *IMA Journal of Mathematical Control and Information*, 3(2-3):61–88, 1986.
- [43] Yuji Nakatsukasa, Olivier Sète, and Lloyd N Trefethen. The aaa algorithm for rational approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, 2018.
- [44] Dirk Deschrijver, Karel Crombecq, Huu Minh Nguyen, and Tom Dhaene. Adaptive sampling algorithm for macromodeling of parameterized  $s$ -parameter responses. *IEEE Transactions on Microwave Theory and Techniques*, 59(1):39–45, 2011.
- [45] Robert Lehmensiek and Petrie Meyer. Creating accurate multivariate rational interpolation models of microwave circuits by using efficient adaptive sampling to minimize the number of computational electromagnetic analyses. *IEEE Transactions on Microwave Theory and Techniques*, 49(8):1419–1430, 2001.
- [46] Elisa Fevola, Alessandro Zanco, Stefano Grivet-Talocia, Tommaso Bradde, and Marco De Stefano. An adaptive sampling process for automated multivariate macromodeling based on hamiltonian-based passivity metrics. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 9(9):1698–1711, 2019.



- 
- [47] F. Ferranti, L. Knockaert, and T. Dhaene. Parameterized S-parameter based macromodeling with guaranteed passivity. *Microwave and Wireless Components Letters, IEEE*, 19(10):608–610, Oct 2009.
- [48] F. Ferranti, L. Knockaert, T. Dhaene, G. Antonini, and D. De Zutter. Parametric macromodeling for tabulated data based on internal passivity. *Microwave and Wireless Components Letters, IEEE*, 20(10):533–535, Oct 2010.
- [49] B. D. O. Anderson and S. Vongpanitlerd. *Network analysis and synthesis*. Prentice-Hall, 1973.
- [50] A. I. Luré. *Some Non-linear Problems in the Theory of Automatic Control: Nekotorye Nelineinye Zadachi Teorii Avtomaticheskogo Regulirovaniya (Gos. Isdat. Tekh. Teor. Lit., 1951, U.S.S.R.) A Translation from the Russian*. H.M. Stationery Office, 1957.
- [51] Krishnan Chemmangat, Francesco Ferranti, Tom Dhaene, and Luc Knockaert. Tree-based sequential sampling algorithm for scalable macromodeling of high-speed systems. In *2012 IEEE 16th Workshop on Signal and Power Integrity (SPI)*, pages 49–52. IEEE, 2012.
- [52] Francesco Ferranti, Krishnan Chemmangat, Tom Dhaene, and Luc Knockaert. Self-constructing compact parametric macromodels for efficient electronic design. In *2013 17th IEEE Workshop on Signal and Power Integrity*, pages 1–4. IEEE, 2013.
- [53] Stefano Grivet-Talocia and Elisa Fevola. Compact parameterized black-box modeling via fourier-rational approximations. *IEEE Transactions on Electromagnetic Compatibility*, 59(4):1133–1142, 2017.
- [54] T. Bradde, S. Grivet-Talocia, M. De Stefano, and A. Zanco. A scalable reduced-order modeling algorithm for the construction of parameterized interconnect macromodels from scattering responses. In *2018 IEEE Symposium on Electromagnetic Compatibility, Signal Integrity and Power Integrity (EMC, SI PI)*, pages 650–655, 2018.
- [55] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter. Macromodeling of multiport systems using a fast implementation of the vector fitting method. *Microwave and Wireless Components Letters, IEEE*, 18(6):383–385, june 2008.
- [56] P. Triverio, M. Nakhla, and S. Grivet-Talocia. Extraction of parametric circuit models from scattering parameters of passive RF components. In *Proc. of the 5th European Microwave Integrated Circuits Conference*, pages 393–396, Paris, September 27 - 28 2010.
- [57] P. Triverio, S. Grivet-Talocia, and A. China. Identification of highly efficient delay-rational macromodels of long interconnects from tabulated

- frequency data. *IEEE Transactions on Microwave Theory and Techniques*, 58(3):566–577, 2010.
- [58] Stefano Grivet-Talocia. A perturbation scheme for passivity verification and enforcement of parameterized macromodels. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 7(11):1869–1881, 2017.
- [59] Muhammad Kabir and Roni Khazaka. Parametric macromodeling of high-speed modules from frequency-domain data using loewner matrix based method. In *2013 IEEE MTT-S International Microwave Symposium Digest (MTT)*, pages 1–4. IEEE, 2013.
- [60] Muhammad Kabir and Roni Khazaka. Fixed-order parametric macromodeling of interconnects from s-parameter data using loewner matrix based method. In *2013 IEEE 22nd Conference on Electrical Performance of Electronic Packaging and Systems*, pages 141–144. IEEE, 2013.
- [61] Antonio Cosmin Ionita and Athanasios C Antoulas. Data-driven parametrized model reduction in the loewner framework. *SIAM Journal on Scientific Computing*, 36(3):A984–A1007, 2014.
- [62] AC Antoulas, AC Ionita, and S Lefteriu. On two-variable rational interpolation. *Linear Algebra and its Applications*, 436(8):2889–2915, 2012.
- [63] Yi Qing Xiao, Stefano Grivet-Talocia, Paolo Manfredi, and Roni Khazaka. A novel framework for parametric loewner matrix interpolation. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 9(12):2404–2417, 2019.
- [64] Andrea Carracedo Rodriguez and Serkan Gugercin. The p-aaa algorithm for data driven modeling of parametric dynamical systems. *arXiv e-prints*, pages arXiv–2003, 2020.
- [65] C. Scherer and S. Weiland. Linear matrix inequalities in control. *Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands*, 2000.
- [66] J. C. Willems. Dissipative dynamical systems part I: General theory. *Archive for Rational Mechanics and Analysis*, 45(5):321–351, 1972.
- [67] M. R. Wohlers and E. J. Beltrami. Distribution theory as the basis of generalized passive-network analysis. *IEEE Trans. on Circuit Theory*, CT-12:164–170, 1965.
- [68] HL Trentelman and JC Willems. Every storage function is a state function. *Systems & Control Letters*, 32(5):249–259, 1997.
- [69] E. S. Kuh and R. A. Rohrer. *Theory of Linear Active Networks*. Holden-Day, 1967.

- [70] M. R. Wohlers. *Lumped and Distributed Passive Networks*. Academic press, 1969.
- [71] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. Canavero, and R. Achar. Stability, causality, and passivity in electrical interconnect models. *IEEE Trans. Advanced Packaging*, 30(4):795–808, Nov 2007.
- [72] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the  $H_\infty$  norm of a transfer matrix and related problems. *Mathematics of Control, Signals and Systems*, 2(3):207–219, 1989.
- [73] Z. Ye, L. M. Silveira, and J. R. Phillips. Fast and reliable passivity assessment and enforcement with extended Hamiltonian pencil. In *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, pages 774–778, Nov 2009.
- [74] R. W. Freund and F. Jarre. An extension of the positive real lemma to descriptor systems. *Optimization methods and software*, 19(1):69–87, 2004.
- [75] C. P. Coelho, J. Phillips, and L. M. Silveira. A convex programming approach for generating guaranteed passive approximations to tabulated frequency-data. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(2):293 – 301, feb. 2004.
- [76] B. Dumitrescu. Parameterization of positive-real transfer functions with fixed poles. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 49(4):523–526, Apr 2002.
- [77] S. Grivet-Talocia and A. Ubolli. A comparative study of passivity enforcement schemes for linear lumped macromodels. *IEEE Trans. Advanced Packaging*, 31(4):673–683, Nov 2008.
- [78] S. Grivet-Talocia. An adaptive sampling technique for passivity characterization and enforcement of large interconnect macromodels. *IEEE Trans. Advanced Packaging*, 30(2):226–237, May 2007.
- [79] D. Saraswat, R. Achar, and M. S. Nakhla. Global passivity enforcement algorithm for macromodels of interconnect subnetworks characterized by tabulated data. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 13(7):819–832, July 2005.
- [80] B. Gustavsen and A. Semlyen. Enforcing passivity for admittance matrices approximated by rational functions. *Power Systems, IEEE Transactions on*, 16(1):97–104, Feb 2001.
- [81] B. Gustavsen. Fast passivity enforcement for pole-residue models by perturbation of residue matrix eigenvalues. *Power Delivery, IEEE Transactions on*, 23(4):2278–2285, Oct 2008.

- [82] Alessandro Zanco, Stefano Grivet-Talocia, Tommaso Bradde, and Marco De Stefano. Enforcing passivity of parameterized lti macromodels via hamiltonian-driven multivariate adaptive sampling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(1):225–238, 2020.
- [83] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys (CSUR)*, 45(3):1–33, 2013.
- [84] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- [85] J. H. Wilkinson. *The algebraic eigenvalue problem*. Clarendon Press, 1965.
- [86] Tommaso Bradde, Stefano Grivet-Talocia, Alessandro Zanco, and Giuseppe C. Calafiore. Data-driven extraction of uniformly stable and passive parameterized macromodels. *IEEE Access*, 10:15786–15804, 2022.
- [87] L. N. Trefethen and D. Bau. *Numerical linear algebra*. Society for Industrial and Applied Mathematics, 1997.
- [88] Lieven Vandenberghe, V Ragu Balakrishnan, Ragnar Wallin, Anders Hansson, and Tae Roh. Interior-point algorithms for semidefinite programming problems derived from the kyp lemma. In *Positive polynomials in control*, pages 195–238. Springer, 2005.
- [89] A. Zanco, S. Grivet-Talocia, T. Bradde, and M. De Stefano. Multivariate macromodeling with stability and passivity constraints. In *2018 IEEE 22nd Workshop on Signal and Power Integrity (SPI)*, pages 1–4, 2018.
- [90] M De Stefano, S Grivet-Talocia, T Bradde, and A Zanco. A framework for the generation of guaranteed stable small-signal bias-dependent behavioral models. In *2018 13th European Microwave Integrated Circuits Conference (EuMIC)*, pages 142–145. IEEE, 2018.
- [91] Alessandro Zanco, Stefano Grivet-Talocia, Tommaso Bradde, and Marco De Stefano. On stabilization of parameterized macromodeling. In *2019 IEEE 23rd Workshop on Signal and Power Integrity (SPI)*, pages 1–4, 2019.
- [92] Bjørn Gustavsen. Fast passivity enforcement for pole-residue models by perturbation of residue matrix eigenvalues. *IEEE Transactions on Power Delivery*, 23(4):2278–2285, 2008.
- [93] Gregory E Fasshauer. *Meshfree approximation methods with MATLAB*, volume 6. World Scientific, 2007.
- [94] Robert Schaback. A practical guide to radial basis functions. *Electronic Resource*, 11:1–12, 2007.

- 
- [95] Holger Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.
- [96] Stefano De Marchi and E Perracchione. Lectures on radial basis functions. *Preprint*, 2018.
- [97] Rolland L Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, 76(8):1905–1915, 1971.
- [98] Robert Schaback. Creating surfaces from scattered data using radial basis functions. *Mathematical methods for curves and surfaces*, 477, 1995.
- [99] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, 4(1):389–396, 1995.
- [100] Zongmin Wu. Compactly supported positive definite radial functions. *Advances in computational mathematics*, 4(1):283–292, 1995.
- [101] Alessandro Zanco, Stefano Grivet-Talocia, Tommaso Bradde, and Marco De Stefano. Uniformly stable parameterized macromodeling through positive definite basis functions. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 10(11):1782–1794, 2020.
- [102] Christopher K Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- [103] Alessandro Zanco and Stefano Grivet-Talocia. Toward fully automated high-dimensional parameterized macromodeling. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(9):1402–1416, 2021.
- [104] Alessandro Zanco and Stefano Grivet-Talocia. High-dimensional parameterized macromodeling with guaranteed stability. In *2019 IEEE 28th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), Montreal (Canada), 6–9 Oct.*, pages 1–3, 2019.
- [105] Gerald Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Academic Press, 1988.
- [106] S. Grivet-Talocia and A. Tabacco. Wavelets on the interval with optimal localization. *Mathematical Models and Methods in Applied Sciences*, 10:441–462, 2000.
- [107] Serge Bernstein. Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités. *Communications de la Société mathématique de Kharkow. 2-ée série*, 13(1):1–2, 1912.

- 
- [108] Salvatore Bernardo Olivadese, Gianni Signorini, Stefano Grivet-Talocia, and Pietro Brenner. Parameterized and dc-compliant small-signal macromodels of rf circuit blocks. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 5(4):508–522, 2015.
- [109] Stefano Grivet-Talocia, Salvatore Bernardo Olivadese, Gianni Signorini, and Pietro Brenner. Compact parameterized macromodels for signal and power integrity analysis of rf and mixed signal systems. In *2013 IEEE Electrical Design of Advanced Packaging Systems Symposium (EDAPS)*, pages 40–43. IEEE, 2013.
- [110] Paolo Manfredi, Dries Vande Ginste, Daniël De Zutter, and Flavio G Canavero. Uncertainty assessment of lossy and dispersive lines in spice-type environments. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 3(7):1252–1258, 2013.
- [111] T Buss. 2ghz low noise amplifier with the bfg425w. Technical report, Philips Semiconductors, B.V., Nijmegen, The Netherlands, 1996.
- [112] HSPICE, Synopsis [online] <http://www.synopsys.com>.
- [113] Michael D McKay, Richard J Beckman, and William J Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.
- [114] Ralph E Carlson and Thomas A Foley. The parameter r2 in multiquadric interpolation. *Computers & Mathematics with Applications*, 21(9):29–42, 1991.
- [115] EJ Kansa and RE Carlson. Improved accuracy of multiquadric interpolation using variable shape parameters. *Computers & Mathematics with Applications*, 24(12):99–120, 1992.
- [116] Stefan Jakobsson, Björn Andersson, and Fredrik Edelvik. Rational radial basis function interpolation with applications to antenna design. *Journal of computational and applied mathematics*, 233(4):889–904, 2009.
- [117] Stefano De Marchi, A Martínez, and Emma Perracchione. Fast and stable rational rbf-based partition of unity interpolation. *Journal of Computational and Applied Mathematics*, 349:331–343, 2019.
- [118] A. Zanco and S. Grivet-Talocia. Hyperparameter determination in multivariate macromodeling based on radial basis functions. In *2020 IEEE 29th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, pages 1–3, 2020.
- [119] Alessandro Zanco and Stefano Grivet-Talocia. A mesh-free adaptive parametric macromodeling strategy with guaranteed stability. In *2020 International Symposium on Electromagnetic Compatibility - EMC EU-ROPE*, pages 1–6, 2020.

- 
- [120] AH-D Cheng. Multiquadric and its shape parameter—a numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. *Engineering analysis with boundary elements*, 36(2):220–239, 2012.
- [121] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.
- [122] Robert Schaback. Multivariate interpolation by polynomials and radial basis functions. *Constructive Approximation*, 21(3):293–317, 2005.
- [123] Gregory E Fasshauer. Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4:21–63, 2011.
- [124] Scott A Sarra and Samuel Cogar. An examination of evaluation algorithms for the rbf method. *Engineering Analysis with Boundary Elements*, 75:36–45, 2017.
- [125] Shmuel Rippa. An algorithm for selecting a good value for the parameter  $c$  in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2-3):193–210, 1999.
- [126] F Marchetti. The extension of rippa’s algorithm beyond loocv. *Applied Mathematics Letters*, 120:107262, 2021.
- [127] S. Elanayar V.T. and Y. C. Shin. Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. *IEEE Transactions on Neural Networks*, 5(4):594–603, 1994.
- [128] W. Kaminski and P. Strumillo. Kernel orthonormalization in radial basis function neural networks. *IEEE Transactions on Neural Networks*, 8(5):1177–1183, 1997.
- [129] David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
- [130] S Chen. Nonlinear time series modelling and prediction using gaussian rbf networks with enhanced clustering and rls learning. *Electronics letters*, 31(2):117–118, 1995.
- [131] W. Pedrycz. Conditional fuzzy clustering in the design of radial basis function neural networks. *IEEE Transactions on Neural Networks*, 9(4):601–612, 1998.

- 
- [132] Ilya A Antonov and VM Saleev. An economic method of computing  $lp_\tau$ -sequences. *USSR Computational Mathematics and Mathematical Physics*, 19(1):252–256, 1979.
  - [133] Marco De Stefano and Stefano Grivet-Talocia. A multivariate adaptive sampling scheme for passivity characterization of parameterized macromodels. In *2021 IEEE 25th Workshop on Signal and Power Integrity (SPI)*, pages 1–3, 2021.
  - [134] Dirk Deschrijver, Karel Crombecq, Huu Minh Nguyen, and Tom Dhaene. Adaptive sampling algorithm for macromodeling of parameterized  $s$ -parameter responses. *IEEE Transactions on Microwave Theory and Techniques*, 59(1):39–45, 2011.
  - [135] Steven George Krantz, Steve Kress, and R Kress. *Handbook of complex variables*. Springer, 1999.
  - [136] AY Alfakih. On the nullspace, the rangespace and the characteristic polynomial of euclidean distance matrices. *Linear algebra and its applications*, 416(2-3):348–354, 2006.
  - [137] Gilbert Strang. The fundamental theorem of linear algebra. *The American Mathematical Monthly*, 100(9):848–855, 1993.
  - [138] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.



# Appendix A

## Proof of Theorem 5

*Proof.* Let us first re-write the denominator  $D(s, \boldsymbol{\vartheta})$  splitting the contributions of real and complex conjugate poles, as

$$D(s, \boldsymbol{\vartheta}) = \underbrace{\sum_{n=0}^{\bar{n}_r} \sum_{\ell_D=1}^{\bar{\ell}_D} \frac{r_{n,\ell}}{s - q_n} \xi_\ell(\boldsymbol{\vartheta})}_{\text{Real poles}} + \underbrace{\sum_{n=0}^{\bar{n}_c} \sum_{\ell_D=1}^{\bar{\ell}_D} \left[ \frac{r'_{n,\ell} + jr''_{n,\ell}}{s - \alpha_n - j\beta_n} + \frac{r'_{n,\ell} - jr''_{n,\ell}}{s - \alpha_n + j\beta_n} \right] \xi_\ell(\boldsymbol{\vartheta})}_{\text{Complex conjugate poles}}. \quad (\text{A.1})$$

We now consider the real and complex conjugate parts separately.

1. *Real poles:* We first consider the contribution of the  $n$ -th pole evaluated for  $s = j\omega$ , which we re-define as

$$D_n^r(j\omega, \boldsymbol{\vartheta}) = \frac{r_n(\boldsymbol{\vartheta})}{j\omega - q_n}, \quad (\text{A.2})$$

with  $q_n \in \mathbb{R}$  and  $q_n < 0$ , and where

$$r_n(\boldsymbol{\vartheta}) = \sum_{\ell=1}^{\bar{\ell}} r_{n,\ell} \xi_\ell(\boldsymbol{\vartheta}). \quad (\text{A.3})$$

Since both  $\xi_\ell(\boldsymbol{\vartheta})$  and  $r_{n,\ell}$  are real-valued, it holds that

$$\Re \{D_n^r(j\omega, \boldsymbol{\vartheta})\} = r_n(\boldsymbol{\vartheta}) \cdot \frac{-q_n}{q_n^2 + \omega^2}. \quad (\text{A.4})$$

Since  $\xi_\ell(\boldsymbol{\vartheta}) \geq 0$ , enforcing the first row in constraints (5.11), for all  $\ell = 1, \dots, \bar{\ell}$ , implies that  $r_n(\boldsymbol{\vartheta}) > 0$ . In turn, since  $q_n < 0$ , this implies

$$\Re \{D_n^r(j\omega, \boldsymbol{\vartheta})\} > 0 \quad \forall \omega, \forall \boldsymbol{\vartheta}. \quad (\text{A.5})$$

2. *Complex conjugate poles:* Let us now consider the  $n$ -th complex conjugate pair in (A.1), restricted to the imaginary axis  $s = j\omega$ , which we re-define as

$$D_n^c(j\omega, \boldsymbol{\vartheta}) = \sum_{\ell=1}^{\bar{\ell}} d_{n,\ell}^c \xi_\ell(\boldsymbol{\vartheta}), \quad (\text{A.6})$$

where

$$d_{n,\ell}^c = \frac{r'_{n,\ell} + jr''_{n,\ell}}{j\omega - \alpha_n - j\beta_n} + \frac{r'_{n,\ell} - jr''_{n,\ell}}{j\omega - \alpha_n + j\beta_n} \quad (\text{A.7})$$

and where  $\alpha_n < 0$ . Therefore, we have

$$\Re \{D_n^c(j\omega, \boldsymbol{\vartheta})\} = \frac{1}{2} \sum_{\ell=1}^{\bar{\ell}} [d_{n,\ell}^c + (d_{n,\ell}^c)^*] \xi_\ell(\boldsymbol{\vartheta}) \quad (\text{A.8})$$

which is nonnegative if

$$d_{n,\ell}^c + (d_{n,\ell}^c)^* > 0. \quad (\text{A.9})$$

Plugging the definition (A.7) in (A.9) leads to the following equivalent condition

$$(\alpha_n^2 + \beta_n^2)(-r'_{n,\ell}\alpha_n + r''_{n,\ell}\beta_n) + \omega^2(-r'_{n,\ell}\alpha_n - r''_{n,\ell}\beta_n) > 0 \quad (\text{A.10})$$

which is verified when the second row in (5.11) holds.

3. *Superposition* Now that we derived positivity conditions on both the real and imaginary parts, combining the individual constraints through (A.1), implies that

$$\Re \{D(j\omega, \boldsymbol{\vartheta})\} > 0 \quad \forall \omega, \forall \boldsymbol{\vartheta} \quad (\text{A.11})$$

since obtained as superposition of nonnegative terms. If we now apply the minimum principle of analytic functions [135], we see that  $\Re \{D(s, \boldsymbol{\vartheta})\} > 0$  for  $\Re \{s\} \geq 0$ , since the minimum real part is attained along  $s = j\omega$ , which is a boundary of the considered domain.



# Appendix B

## Proof of Theorem 6

### B.1 Euclidean Distance Matrices

Let us begin with two preliminary results about Euclidean Distance Matrices (EDM), which form a basis for the main developments. The following Lemma states useful results [136] on rank and column space of square EDMs.

**Lemma 4.** *Consider a set of known vectors  $\{\boldsymbol{\vartheta}_m \in \mathbb{R}^\rho, m = 1 \dots, \bar{m}\}$ , and define the square Euclidean Distance Matrix  $\bar{\mathbf{D}} \in \mathbb{R}^{\bar{m} \times \bar{m}}$  whose  $(i, j)$ -th element is  $\|\boldsymbol{\vartheta}_i - \boldsymbol{\vartheta}_j\|^2$ . Then,*

1. *there exists a unique vector  $\mathbf{p} \in \mathbb{R}^{\bar{m}}$  such that*

$$\bar{\mathbf{D}} \mathbf{p} = \mathbf{1}_{\bar{m}} \tag{B.1}$$

2.  $\text{rank}(\bar{\mathbf{D}}) \leq \rho + 2$ .

*Proof.* See [136]. □

The following extends the result of Lemma 4 to non-square EDMs.

**Lemma 5.** *Consider a set of known vectors  $\{\boldsymbol{\vartheta}_m \in \mathbb{R}^\rho, m = 1, \dots, \bar{m}\}$  and assume  $\rho + 2 < \bar{\ell} < \bar{m}$ . Define the rectangular Euclidean Distance Matrix  $\mathbf{D}_1 \in \mathbb{R}^{\bar{m} \times \bar{\ell}}$ , whose  $(m, \ell)$ -th element is  $\|\boldsymbol{\vartheta}_m - \boldsymbol{\vartheta}_\ell\|^2$ . Then,*

1.  $\text{rank}(\mathbf{D}_1) \leq \rho + 2$ .

2. denoting as  $\mathcal{V}_1$  the null space of  $\mathbf{D}_1$ , it holds that  $\mathcal{V}_1 \neq \emptyset$  and  $\mathcal{V}_1 \perp \mathbf{1}_{\bar{\ell}}$ .

*Proof.* Let us consider a square EDM  $\bar{\mathbf{D}} \in \mathbb{R}^{\bar{m} \times \bar{m}}$  generated by vectors  $\boldsymbol{\vartheta}_m$ . From Lemma 4 we know that  $\text{rank}(\bar{\mathbf{D}}) \leq \rho + 2$ . A ‘‘tall and thin’’ rectangular matrix  $\mathbf{D}_1 \in \mathbb{R}^{\bar{m} \times \bar{\ell}}$  is obtained from  $\bar{\mathbf{D}}$  by deleting  $\bar{m} - \bar{\ell}$  columns. Therefore, we conclude that its rank cannot exceed the rank of  $\bar{\mathbf{D}}$ . Hence, we conclude that

$$\text{rank}(\mathbf{D}_1) \leq \rho + 2 \quad (\text{B.2})$$

which proves 1).

Under the assumption that  $\bar{\ell} > \rho + 2$ , the columns of  $\mathbf{D}_1$  are linearly dependent. Therefore, its null space  $\mathcal{V}_1$  is non-empty and

$$\dim \mathcal{V}_1 \geq \bar{\ell} - \rho - 2. \quad (\text{B.3})$$

If we apply Lemma 4 to  $\bar{\mathbf{D}}$  and we delete the last  $\bar{m} - \bar{\ell}$  rows from (B.1), we prove that there exists a vector  $\mathbf{p} \in \mathbb{R}^{\bar{m}}$  such that  $\mathbf{D}_1^\top \mathbf{p} = \mathbf{1}_{\bar{\ell}}$ . Hence, the Fundamental Theorem of Linear Algebra [137] states that

$$\mathcal{V}_1 \perp \mathbf{1}_{\bar{\ell}} \quad (\text{B.4})$$

which proves 2). □

## B.2 Least Singular Values of Gaussian Kernel Matrices

Using now Lemma 5, we can prove that the least singular value of a non-square Gaussian kernel matrix  $\mathbf{K}(\varepsilon)$  (without loss of generality, we will drop the subscripts  $(N, D)$  when referring to  $\mathbf{K}(\varepsilon)$ ) decays as an integer power of  $\varepsilon$  as the shape parameter approaches 0. This is the first main step to provide a theoretical ground to the proposed Algorithm 2. We have the following

**Theorem 7.** *Consider a Gaussian Kernel Matrix  $\mathbf{K}(\varepsilon)$  as defined in (5.9), with  $\bar{m} > \bar{\ell} > \rho + 2$ . For  $\varepsilon \rightarrow 0$ , the least singular value  $\underline{\sigma}_{\mathbf{K}}(\varepsilon)$  decays asymptotically as*

$$\underline{\sigma}_{\mathbf{K}}(\varepsilon) = O(\varepsilon^\nu) \quad \text{as } \varepsilon \rightarrow 0, \quad \nu \in \mathbb{N}, \quad \nu \geq 4. \quad (\text{B.5})$$

*Proof.* Let us consider the EDM  $\mathbf{D}_1 \in \mathbb{R}^{\bar{m} \times \bar{\ell}}$ , whose  $(m, \ell)$ -th element is  $\|\boldsymbol{\vartheta}_m - \boldsymbol{\vartheta}_\ell\|^2$ . The kernel matrix  $\mathbf{K}(\varepsilon)$  defined in (5.9) is obtained from  $\mathbf{D}_1$  by multiplying each element by  $-\varepsilon^2$  and taking the exponential. To better explore the asymptotic limit  $\varepsilon \rightarrow 0$ , we form the elementwise MacLaurin series expansion, as

$$\mathbf{K}(\varepsilon) = \sum_{p=0}^{\infty} \frac{(-1)^p}{p!} \varepsilon^{2p} \mathbf{D}_p \quad (\text{B.6})$$

where  $\mathbf{D}_p$  denotes the  $p$ -th *Hadamard power* of  $\mathbf{D}_1$ , i.e.  $(\mathbf{D}_p)_{m,\ell} = \|\boldsymbol{\vartheta}_m - \boldsymbol{\vartheta}_\ell\|^{2p}$ .

Following its variational characterization, the least singular value  $\underline{\sigma}_{\mathbf{K}}(\varepsilon)$  of  $\mathbf{K}(\varepsilon)$  is defined as

$$\underline{\sigma}_{\mathbf{K}}(\varepsilon) = \min_{\|\mathbf{z}\|=1} \|\mathbf{K}(\varepsilon)\mathbf{z}\|. \quad (\text{B.7})$$

Now, plugging (B.6) in (B.7) we have

$$\underline{\sigma}_{\mathbf{K}}(\varepsilon) = \min_{\|\mathbf{z}\|=1} \left\| \sum_{p=0}^{\infty} \frac{(-1)^p}{p!} \varepsilon^{2p} \mathbf{D}_p \mathbf{z} \right\|, \quad (\text{B.8})$$

Hence, the leading power  $\nu$  of  $\underline{\sigma}_{\mathbf{K}}(\varepsilon)$  for  $\varepsilon \rightarrow 0$  equals the first term  $\nu = 2p$  in (B.8) for which the matrix-vector product  $\mathbf{D}_p \mathbf{z}$  does not vanish identically, independently on the choice of the vector  $\mathbf{z}$ . To better characterize this information, let us analyze the null spaces  $\mathcal{V}_p$  of matrices  $\mathbf{D}_p$  for increasing  $p$ .

- $p = 0$ . The null space  $\mathcal{V}_0$  can be easily characterized noting that  $\mathbf{D}_0 = \mathbf{1}_{\bar{m} \times \bar{\ell}}$  is the matrix of all ones. Therefore, the null space  $\mathcal{V}_0$  contains all the vectors  $\mathbf{z} \in \mathbb{R}^{\bar{\ell}}$  that are orthogonal to  $\mathbf{1}_{\bar{\ell}}$ , i.e.

$$\mathcal{V}_0 = \{\mathbf{z} \in \mathbb{R}^{\bar{\ell}} : \mathbf{z}^T \cdot \mathbf{1}_{\bar{\ell}} = 0\}. \quad (\text{B.9})$$

For any  $\mathbf{z} \in \mathcal{V}_0$ ,  $\|\mathbf{z}\| = 1$ , it holds that  $\mathbf{D}_p \mathbf{z} = \mathbf{0}$ , proving that the least singular value  $\underline{\sigma}_{\mathbf{K}}(\varepsilon) \rightarrow 0$  for  $\varepsilon \rightarrow 0$  with a leading power at least  $\nu = 2$ .

- $p = 1$ . By assumption  $\bar{\ell} > \rho + 2$ , thus Lemma 5 ensures that the null space  $\mathcal{V}_1$  of  $\mathbf{D}_1$  is non-empty and has dimension larger than  $\bar{\ell} - \rho - 2$ . In addition, Lemma 5 states that  $\mathcal{V}_1$  is orthogonal to the vector  $\mathbf{1}_{\bar{\ell}}$ . Recalling the definition (B.9) of  $\mathcal{V}_0$ , we can conclude that  $\mathcal{V}_1 \subseteq \mathcal{V}_0$  and that  $\mathcal{V}_1 \cap \mathcal{V}_0 = \mathcal{V}_1 \neq \emptyset$ . Thus, it always exists a non-vanishing vector

$\mathbf{z} \in \mathcal{V}_1$  that cancels both the terms  $\mathbf{D}_1 \mathbf{z}$  and  $\mathbf{D}_0 \mathbf{z}$  in (B.8). This proves that the leading power of  $\underline{\sigma}_{\mathbf{K}}(\varepsilon)$  for  $\varepsilon \rightarrow 0$  is at least  $\nu = 4$ .

- $p = 2$ . Unfortunately, an explicit characterization of the null space  $\mathcal{V}_2$  of matrix  $\mathbf{D}_2$  is not available. So, we can not proceed further with our analysis.

We conclude that the scaling power for the least singular value  $\underline{\sigma}_{\mathbf{K}}(\varepsilon)$  for  $\varepsilon \rightarrow 0$  is at least  $\nu \geq 4$ .  $\square$

### B.3 From Kernel to Regressor Matrices

Given the results of Theorem 7, we now prove that the same scaling power propagates to all the PSK matrices.

We proceed in steps, following a constructive approach based on the definitions of Section 5.2.

First, we consider the matrix  $\mathbf{X}(\varepsilon)$ , defined as in 5.8. A standard result states that the singular value spectrum of the Kronecker product  $\mathbf{\Phi} \otimes \mathbf{K}(\varepsilon)$  is obtained as the product of all the individual singular value spectra product of the two matrices  $\mathbf{\Phi}$  and  $\mathbf{K}(\varepsilon)$ . Thus, the least singular value  $\underline{\sigma}_{\mathbf{X}(\varepsilon)} = \underline{\sigma}_{\mathbf{\Phi} \otimes \mathbf{K}(\varepsilon)}$  is the product of the least singular values of both matrices

$$\underline{\sigma}_{\mathbf{\Phi} \otimes \mathbf{K}(\varepsilon)} = \underline{\sigma}_{\mathbf{K}(\varepsilon)} \cdot \underline{\sigma}_{\mathbf{\Phi}} \quad (\text{B.10})$$

where  $\underline{\sigma}_{\mathbf{\Phi}}$  does not depend on  $\varepsilon$ . In turn, this implies that

$$\underline{\sigma}_{\mathbf{\Phi} \otimes \mathbf{K}(\varepsilon)} = O(\varepsilon^\nu) \quad \text{as } \varepsilon \rightarrow 0, \quad (\text{B.11})$$

with the same  $\nu \geq 4$  that holds for  $\underline{\sigma}_{\mathbf{K}(\varepsilon)}$ .

We can now proceed with matrices  $\mathbf{\Gamma}^\mu(\varepsilon)$  and  $\mathbf{\Xi}_{(i,j)}^\mu(\varepsilon)$ . These matrices share the same basic structure, addressed in the following

**Lemma 6.** *Let  $\mathbf{\Phi}$  and  $\mathbf{K}(\varepsilon)$  be defined as in (2.28) and (5.9). Assuming  $\Delta$  to be a generic real-valued non-singular square matrix of compatible size, for  $\varepsilon \rightarrow 0$  the least singular value of  $\Delta \cdot (\mathbf{\Phi} \otimes \mathbf{K}(\varepsilon))$  decays as  $\varepsilon^\nu$ ,  $\nu \in \mathbb{N}$ ,  $\nu \geq 4$ .*

*Proof.* For any fixed instance of  $\varepsilon$ , we denote the SVD of  $\Phi \otimes \mathbf{K}(\varepsilon)$  as  $\mathbf{U}(\varepsilon)\Sigma(\varepsilon)\mathbf{V}^\top(\varepsilon)$ . In addition, we denote the singular values, collected in descending order along the diagonal of matrix  $\Sigma(\varepsilon)$ , as  $\sigma_t(\varepsilon)$ ,  $t = 1, \dots, \bar{t}$ , and  $\mathbf{U}^\top(\varepsilon)\mathbf{U}(\varepsilon) = \mathbf{I}$ ,  $\mathbf{V}^\top(\varepsilon)\mathbf{V}(\varepsilon) = \mathbf{I}$ . From (B.11) we know that  $\sigma_{\bar{t}}(\varepsilon) = \underline{\sigma}_{\Phi \otimes \mathbf{K}}(\varepsilon)$  decays as  $O(\varepsilon^\nu)$ , with  $\nu \geq 4$ , for  $\varepsilon \rightarrow 0$ .

We start considering the following identity

$$\Delta \cdot (\Phi \otimes \mathbf{K}(\varepsilon)) = \Delta \cdot \mathbf{U}(\varepsilon)\Sigma(\varepsilon)\mathbf{V}^\top(\varepsilon) \quad (\text{B.12})$$

and we compute the QR decomposition  $\Delta \cdot \mathbf{U}(\varepsilon) = \mathbf{Q}(\varepsilon)\mathbf{R}(\varepsilon)$ , where the matrix  $\mathbf{R}(\varepsilon) \in \mathbb{R}^{\bar{t} \times \bar{t}}$  is upper triangular and  $\mathbf{Q}^\top(\varepsilon)\mathbf{Q}(\varepsilon) = \mathbf{I}$ . We thus have

$$\Delta \cdot (\Phi \otimes \mathbf{K}(\varepsilon)) = \mathbf{Q}(\varepsilon)\mathbf{R}(\varepsilon)\Sigma(\varepsilon)\mathbf{V}(\varepsilon)^\top. \quad (\text{B.13})$$

Let us additionally denote the columns of  $\mathbf{R}(\varepsilon)$  as  $\mathbf{r}_t$ , i.e.  $\mathbf{R}(\varepsilon) = (\mathbf{r}_1 \ \dots \ \mathbf{r}_{\bar{t}})$ . By noting that  $\Delta$  is non-singular and that the columns of  $\mathbf{U}(\varepsilon)$  are mutually orthogonal, we can conclude that that  $\mathbf{R}(\varepsilon)$  is square and invertible and its columns  $\mathbf{r}_t$  are linearly independent.

Now, we evaluate the SVD of the product  $\mathbf{R}(\varepsilon)\Sigma(\varepsilon) = \hat{\mathbf{U}}(\varepsilon)\hat{\Sigma}(\varepsilon)\hat{\mathbf{V}}^\top(\varepsilon)$ , whose singular values are defined as  $\hat{\sigma}_t(\varepsilon)$ , for  $t = 1, \dots, \bar{t}$  and  $\hat{\mathbf{U}}^\top(\varepsilon)\hat{\mathbf{U}}(\varepsilon) = \mathbf{I}$ ,  $\hat{\mathbf{V}}^\top(\varepsilon)\hat{\mathbf{V}}(\varepsilon) = \mathbf{I}$ . Plugging this result in (B.13) leads to

$$\Delta \cdot (\Phi \otimes \mathbf{K}(\varepsilon)) = \tilde{\mathbf{U}}(\varepsilon)\hat{\Sigma}(\varepsilon)\tilde{\mathbf{V}}^\top(\varepsilon) \quad (\text{B.14})$$

where the columns of matrices  $\tilde{\mathbf{U}}(\varepsilon) = \mathbf{Q}(\varepsilon)\hat{\mathbf{U}}(\varepsilon)$  and  $\tilde{\mathbf{V}}(\varepsilon) = \mathbf{V}(\varepsilon)\hat{\mathbf{V}}(\varepsilon)$  are orthogonal by definition. We immediately notice that (B.14) provides a singular value decomposition for our initial product  $\Delta \cdot (\Phi \otimes \mathbf{K}(\varepsilon))$ , that is unique up to permutations of the singular values that, as usual, are assumed to be sorted in descending order.

From (B.14) we notice that  $\mathbf{R}(\varepsilon)\Sigma(\varepsilon)$  and  $\Delta \cdot (\Phi \otimes \mathbf{K}(\varepsilon))$  share the same singular values. There, we can infer the asymptotic behavior of the least singular value of  $\Delta \cdot (\Phi \otimes \mathbf{K}(\varepsilon))$  by looking at the least singular value  $\hat{\sigma}_{\bar{t}}(\varepsilon)$  of  $\mathbf{R}(\varepsilon)\Sigma(\varepsilon)$ , characterized in a variational form as

$$\hat{\sigma}_{\bar{t}}(\varepsilon) = \min_{\|\mathbf{w}\|=1} \|\mathbf{R}(\varepsilon)\Sigma(\varepsilon)\mathbf{w}\|. \quad (\text{B.15})$$



We now define the vector  $\mathbf{w}_{\bar{t}} = (0, \dots, 0, 1)^\top$ , so that any unit norm vector  $\mathbf{w}$  can be written as

$$\mathbf{w} = \alpha \mathbf{w}_{\bar{t}} + \beta \mathbf{v}, \quad \mathbf{v} \perp \mathbf{w}_{\bar{t}}, \quad \|\mathbf{v}\| = 1, \quad \alpha^2 + \beta^2 = 1. \quad (\text{B.16})$$

The matrix product  $\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}$  thus reads

$$\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w} = \alpha \mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}_{\bar{t}} + \beta \mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{v} \quad (\text{B.17})$$

Since by assumption  $\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}_{\bar{t}} = \sigma_{\bar{t}}(\varepsilon)\mathbf{r}_{\bar{t}}(\varepsilon)$ , we write

$$\|\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}\| = \left\| \alpha \sigma_{\bar{t}}(\varepsilon)\mathbf{r}_{\bar{t}} + \beta \sum_{t < \bar{t}} \mathbf{r}_t \sigma_t(\varepsilon) v_t \right\| \quad (\text{B.18})$$

where  $v_t$  is the  $t$ -th component of vector  $\mathbf{v}$ . We know from (B.11) that, for  $\varepsilon \rightarrow 0$ , the least singular value  $\sigma_{\bar{t}}(\varepsilon)$  decays as  $O(\varepsilon^\nu)$ ,  $\nu \geq 4$ ; in addition, any other singular value  $\sigma_t(\varepsilon)$  for  $t < \bar{t}$  cannot decay faster than  $\sigma_{\bar{t}}(\varepsilon)$ . We have only two possible cases

1.  $\sigma_t(\varepsilon) = O(\varepsilon^\nu)$ ,  $\forall t$

In this case, we have that  $\|\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}\| = O(\varepsilon^\nu)$  for any  $\mathbf{w}$  defined as in (B.16), since all the terms in (B.18) share the same scaling power for  $\varepsilon \rightarrow 0$ .

2.  $\sigma_t(\varepsilon) = O(\varepsilon^\gamma)$ , with  $\gamma < \nu$  for some  $t$

In this case, instead, there exists  $\mathbf{w}_t$  for which  $\eta_t(\varepsilon) = \|\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}_t\| = O(\varepsilon^\gamma)$  (Which can be easily found as an all-zero vector with a single unit entry at component  $t$ ). For such a vector, the term  $\eta_t(\varepsilon) = \|\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}_t\|$  decays slower than  $\eta_{\bar{t}}(\varepsilon) = \|\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}_{\bar{t}}\|$  for  $\varepsilon \rightarrow 0$ . However, although for some  $\varepsilon$  it may happen that  $\eta_t(\varepsilon) < \eta_{\bar{t}}(\varepsilon)$ , there exists  $\varepsilon^*$  so that  $\eta_{\bar{t}}(\varepsilon) < \eta_t(\varepsilon)$  for  $\varepsilon < \varepsilon^*$ . We can conclude that, for  $\varepsilon \rightarrow 0$ , the vector  $\mathbf{w} = \mathbf{w}_{\bar{t}}$  leads to the smallest value of  $\eta(\varepsilon)$ .

These considerations lead us to conclude that

$$\min_{\|\mathbf{w}\|=1} \|\mathbf{R}(\varepsilon)\boldsymbol{\Sigma}(\varepsilon)\mathbf{w}\| = \underline{\sigma}\{\boldsymbol{\Delta} \cdot (\boldsymbol{\Phi} \otimes \mathbf{K}(\varepsilon))\} = O(\varepsilon^\nu) \quad (\text{B.19})$$

with  $\nu \geq 4$ . □

Finally, we recall that the PSK matrices  $\mathbf{\Gamma}^\mu(\varepsilon)$  and  $\mathbf{\Xi}_{(i,j)}^\mu(\varepsilon)$  in (5.7) are built from the kernel matrices  $\mathbf{K}_N(\varepsilon)$  and  $\mathbf{K}_D(\varepsilon)$  respectively, which may be different depending on the selected model orders. Thus, assuming that for  $\varepsilon \rightarrow 0$  the associated least singular values decay as  $\underline{\sigma}_{\mathbf{K}_N}(\varepsilon) = O(\varepsilon^{\nu_N})$  and  $\underline{\sigma}_{\mathbf{K}_D}(\varepsilon) = O(\varepsilon^{\nu_D})$ , Lemma 6 ensures that

$$\underline{\sigma}_{\mathbf{\Gamma}}(\varepsilon) = O(\varepsilon^{\nu_N}), \quad \underline{\sigma}_{\mathbf{\Xi}_{(i,j)}}(\varepsilon) = O(\varepsilon^{\nu_D}) \quad (\text{B.20})$$

as  $\varepsilon \rightarrow 0$ , with  $\nu_D \geq 4$ ,  $\nu_N \geq 4$ .

## B.4 Assembling regressor matrices

Now we are ready to derive an asymptotic estimate for the singular value  $\underline{\sigma}_{\mathbf{\Psi}}(\varepsilon)$  of the regressor  $\mathbf{\Psi}^\mu(\varepsilon)$  in (5.6). To this end, let us define two auxiliary matrices

$$\mathbf{\Psi}_L^\mu(\varepsilon) = \begin{pmatrix} \mathbf{\Gamma}^\mu(\varepsilon) & & \\ & \ddots & \\ & & \mathbf{\Gamma}^\mu(\varepsilon) \end{pmatrix}, \quad \mathbf{\Psi}_R^\mu(\varepsilon) = \begin{pmatrix} \mathbf{\Xi}_{(1,1)}^\mu(\varepsilon) \\ \vdots \\ \mathbf{\Xi}_{(P,P)}^\mu(\varepsilon) \end{pmatrix} \quad (\text{B.21})$$

so that

$$\mathbf{\Psi}^\mu(\varepsilon) = \begin{pmatrix} \mathbf{\Psi}_L^\mu(\varepsilon) & \mathbf{\Psi}_R^\mu(\varepsilon) \end{pmatrix} \quad (\text{B.22})$$

and let us study  $\mathbf{\Psi}_L^\mu(\varepsilon)$  and  $\mathbf{\Psi}_R^\mu(\varepsilon)$  separately.

First, it is trivial to see that the singular values of  $\mathbf{\Psi}_L^\mu(\varepsilon)$  are the same of  $\mathbf{\Gamma}^\mu(\varepsilon)$ , replicated  $P^2$  times. Hence, it is also easy to see that  $\underline{\sigma}_{\mathbf{\Psi}_L}(\varepsilon) = O(\varepsilon^{\nu_N})$  as  $\varepsilon \rightarrow 0$ .

In case of matrix  $\mathbf{\Psi}_R^\mu(\varepsilon)$ , we can prove the following result

**Lemma 7.** *Let  $\mathbf{\Psi}_R^\mu(\varepsilon)$  be defined as in (B.21). Assuming that for all  $i, j = 1, \dots, P$  the least singular value of each block  $\underline{\sigma}_{\mathbf{\Xi}_{(i,j)}^\mu(\varepsilon)} = O(\varepsilon^{\nu_D})$  as  $\varepsilon \rightarrow 0$ , then also*

$$\underline{\sigma}_{\mathbf{\Psi}_R}(\varepsilon) = O(\varepsilon^{\nu_D}), \quad \varepsilon \rightarrow 0. \quad (\text{B.23})$$

*Proof.* We recall that

$$\mathbf{\Xi}_{(i,j)}(\varepsilon) = -\check{\mathbf{H}}_{(i,j)} \mathbf{W}^{\mu-1} \mathbf{X}_D(\varepsilon) \quad (\text{B.24})$$

For the further developments, let us denote the SVD of  $\mathbf{X}_D(\varepsilon)$  as  $\mathbf{U}(\varepsilon)\mathbf{\Sigma}(\varepsilon)\mathbf{V}^\top(\varepsilon)$ . We denote the singular values (collected in descending order along the diagonal of  $\mathbf{\Sigma}(\varepsilon)$ ) as  $\sigma_t(\varepsilon)$ ,  $t = 1, \dots, \bar{t}$ , and the following relations hold  $\mathbf{U}^\top(\varepsilon)\mathbf{U}(\varepsilon) = \mathbf{I}$ ,  $\mathbf{V}^\top(\varepsilon)\mathbf{V}(\varepsilon) = \mathbf{I}$ . We thus write

$$\mathbf{\Psi}_R^\mu(\varepsilon) = \underbrace{\begin{pmatrix} -\check{\mathbf{H}}_{(1,1)} \mathbf{W}^{\mu-1} \mathbf{U}(\varepsilon) \\ \vdots \\ -\check{\mathbf{H}}_{(P,P)} \mathbf{W}^{\mu-1} \mathbf{U}(\varepsilon) \end{pmatrix}}_{\mathbf{L}(\varepsilon)} \mathbf{\Sigma}(\varepsilon) \mathbf{V}^\top(\varepsilon) \quad (\text{B.25})$$

which shares with (B.12) the same structure, obtained replacing  $\mathbf{\Delta} \cdot \mathbf{U}(\varepsilon)$  with  $\mathbf{L}(\varepsilon)$ .

Following the same procedure as in the proof of Lemma 6 we conclude that

$$\underline{\sigma}_{\mathbf{\Psi}_R}(\varepsilon) = O(\varepsilon^{\nu_D}) \quad (\text{B.26})$$

as  $\varepsilon \rightarrow 0$ . □

We can state our final result

**Theorem 8.** *Let  $\mathbf{\Psi}^\mu(\varepsilon)$  be defined as in (B.22). If for  $\varepsilon \rightarrow 0$  it holds that  $\underline{\sigma}_{\mathbf{\Psi}_L}(\varepsilon) = O(\varepsilon^{\nu_N})$  and  $\underline{\sigma}_{\mathbf{\Psi}_R}(\varepsilon) = O(\varepsilon^{\nu_D})$ , then*

$$\underline{\sigma}_{\mathbf{\Psi}}(\varepsilon) = O(\varepsilon^\tau), \quad \tau = \max\{\nu_N, \nu_D\}. \quad (\text{B.27})$$

*Proof.* As usual, we characterize the least singular value as

$$\underline{\sigma}_{\mathbf{\Psi}}(\varepsilon) = \min_{\|\mathbf{w}\|=1} \|\mathbf{\Psi}^\mu(\varepsilon)\mathbf{w}\|. \quad (\text{B.28})$$

Moreover, we split the vector  $\mathbf{w}$  as

$$\mathbf{w} = \begin{pmatrix} \alpha \mathbf{w}_L \\ \beta \mathbf{w}_R \end{pmatrix} \quad (\text{B.29})$$

where  $\|\mathbf{w}_L\| = \|\mathbf{w}_R\| = 1$  and  $\alpha^2 + \beta^2 = 1$  to ensure  $\|\mathbf{w}\| = 1$ .

Defining the following SVDs

$$\begin{aligned}\Psi_L^\mu(\varepsilon) &= \mathbf{U}_L(\varepsilon)\Sigma_L(\varepsilon)\mathbf{V}_L^\top(\varepsilon) \\ \Psi_R^\mu(\varepsilon) &= \mathbf{U}_R(\varepsilon)\Sigma_R(\varepsilon)\mathbf{V}_R^\top(\varepsilon)\end{aligned}\tag{B.30}$$

it holds that

$$\begin{aligned}\Psi^\mu(\varepsilon)\mathbf{w} &= \alpha\mathbf{U}_L(\varepsilon)\Sigma_L(\varepsilon)\mathbf{V}_L^\top(\varepsilon)\mathbf{w}_L \\ &\quad + \beta\mathbf{U}_R(\varepsilon)\Sigma_R(\varepsilon)\mathbf{V}_R^\top(\varepsilon)\mathbf{w}_R\end{aligned}\tag{B.31}$$

For any fixed instance of  $\varepsilon$ , we consider the first summation term of (B.31), and we pick  $\mathbf{w}_L = \underline{\mathbf{v}}_L$ , where  $\underline{\mathbf{v}}_L$  is defined as the last column of  $\mathbf{V}_L(\varepsilon)$ . Hence, it trivially follows that  $\mathbf{V}_L^\top(\varepsilon)\underline{\mathbf{v}}_L = (0, \dots, 1)^\top$ . We proceed recalling that the matrix  $\Sigma_L(\varepsilon)$  stores the singular values of  $\Psi_L^\mu(\varepsilon)$  sorted in descending order, thus  $\Sigma_L(\varepsilon)\mathbf{V}_L^\top(\varepsilon)\underline{\mathbf{v}}_L = (0, \dots, \sigma_{\Psi_L}(\varepsilon))^\top$  and the first summation term in (B.31) reduces to

$$\alpha\mathbf{U}_L(\varepsilon)\Sigma_L(\varepsilon)\mathbf{V}_L^\top(\varepsilon)\underline{\mathbf{v}}_L = \alpha\underline{\mathbf{u}}_L\sigma_{\Psi_L}(\varepsilon)\tag{B.32}$$

where  $\underline{\mathbf{u}}_L$  denotes the last column of  $\mathbf{U}_L(\varepsilon)$ .

We repeat the same procedure on the second term of (B.31), choosing the vector  $\mathbf{w}_R = \underline{\mathbf{v}}_R$  as the last column of  $\mathbf{V}_R(\varepsilon)$ , obtaining

$$\beta\mathbf{U}_R(\varepsilon)\Sigma_R(\varepsilon)\mathbf{V}_R^\top(\varepsilon)\underline{\mathbf{v}}_R = \beta\underline{\mathbf{u}}_R\sigma_{\Psi_R}(\varepsilon)\tag{B.33}$$

where  $\underline{\mathbf{u}}_R$  is the last column of  $\mathbf{U}_R(\varepsilon)$ .

For this choice of  $\mathbf{w}$  we thus have

$$\mathbf{w} = \begin{pmatrix} \alpha\underline{\mathbf{v}}_L \\ \beta\underline{\mathbf{v}}_R \end{pmatrix} \rightarrow \Psi^\mu(\varepsilon)\mathbf{w} = \alpha\underline{\mathbf{u}}_L\sigma_{\Psi_L}(\varepsilon) + \beta\underline{\mathbf{u}}_R\sigma_{\Psi_R}(\varepsilon)\tag{B.34}$$

where all vectors  $\mathbf{w}$ ,  $\underline{\mathbf{v}}_{L,R}$ ,  $\underline{\mathbf{u}}_{L,R}$  have unit length and the least singular values  $\sigma_{\Psi_{L,R}}(\varepsilon)$  scale as  $O(\varepsilon^{\nu_{N,D}})$  for  $\varepsilon \rightarrow 0$ , by assumption. When  $\mathbf{w}$  is chosen as in (B.34), we therefore have

$$\min_{\|\mathbf{w}\|=1} \left\| \alpha\underline{\mathbf{u}}_L\sigma_{\Psi_L}(\varepsilon) + \beta\underline{\mathbf{u}}_R\sigma_{\Psi_R}(\varepsilon) \right\| = O(\varepsilon^\tau)\tag{B.35}$$

with  $\tau = \max \{\nu_N, \nu_D\}$ , which is attained by selecting  $\alpha = 1, 0$  and  $\beta = 0, 1$  respectively.

Following similar arguments as in the proof of Lemma 6, we prove that  $\mathbf{w}$  as defined in (B.34) the largest scaling exponent  $\tau$  for  $\varepsilon \rightarrow 0$ .

Hence, the minimum is achieved by (B.35) and we can conclude that

$$\underline{\sigma}_{\Psi}(\varepsilon) = \min_{\|\mathbf{w}\|=1} \|\Psi^\mu(\varepsilon)\mathbf{w}\| = O(\varepsilon^\tau) \quad (\text{B.36})$$

with  $\tau = \max \{\nu_N, \nu_D\}$ . □

# Appendix C

## Proof of Proposition 1

In the following, we are going to prove Proposition 1 of Section 6.2.2.

*Proof.* Consider the Gaussian Kernel Matrix  $\mathbf{K}(\varepsilon)$  defined in (5.9), and let us re-write it in terms of its column components, as

$$\mathbf{K}(\varepsilon) = [\mathbf{k}_1(\varepsilon), \dots, \mathbf{k}_{\bar{\ell}}(\varepsilon)], \quad (\text{C.1})$$

where

$$\mathbf{k}_\ell(\varepsilon) = \left[ e^{-\varepsilon^2 \|\boldsymbol{\vartheta}_1 - \boldsymbol{\vartheta}_\ell\|^2}, \dots, e^{-\varepsilon^2 \|\boldsymbol{\vartheta}_{\bar{m}} - \boldsymbol{\vartheta}_\ell\|^2} \right]^\top \quad (\text{C.2})$$

For the further developments, it is worth noting that each element of  $\mathbf{k}_\ell$  is bounded in the interval  $[0, 1]$  for  $0 \leq \varepsilon \leq \infty$ . In addition, under the assumption that the RBFs are centered on the available training samples, there exists unique  $\boldsymbol{\vartheta}_m = \boldsymbol{\vartheta}_\ell$ ,  $\ell = 1, \dots, \bar{\ell}$ . Therefore, independently of  $\varepsilon$ , each vector  $\mathbf{k}_\ell(\varepsilon)$  includes a single element which is equal to 1.

For  $\varepsilon \in \{\mathbb{R}^+ \cup \infty\}$ , the vectors  $\mathbf{k}_\ell(\varepsilon)$  thus range from the unitary vector  $\mathbf{1} = [1, 1, \dots, 1]^\top$  (for  $\varepsilon \rightarrow 0$ ) to the canonical basis  $\mathbf{e}_\ell = [0, \dots, 1, \dots, 0]^\top$  (for  $\varepsilon \rightarrow \infty$ ). We can conclude that

$$\|\mathbf{k}_\ell(\varepsilon)\|_2 \in [1, \sqrt{\bar{m}}]. \quad (\text{C.3})$$

and

$$\|\mathbf{K}(\varepsilon)\| \in [1, \sqrt{\bar{m}\bar{\ell}}] \quad (\text{C.4})$$

In our proof, we will consider the upper and lower bounds separately.

## C.1 Proving the upper bound

Following its variational definition, the leading singular value  $\bar{\sigma}_{\mathbf{K}}(\varepsilon)$  reads

$$\bar{\sigma}_{\mathbf{K}}(\varepsilon) = \sup_{\|\mathbf{z}\|=1} \|\mathbf{K}(\varepsilon)\mathbf{z}\|. \quad (\text{C.5})$$

In order to prove that  $\bar{\sigma}_{\mathbf{K}}(\varepsilon) \leq \sqrt{\bar{m}\bar{\ell}}$ ,  $\forall \varepsilon$ , let us write

$$\|\mathbf{K}(\varepsilon)\mathbf{z}\| = \|\mathbf{k}_1(\varepsilon)z_1 + \dots + \mathbf{k}_{\bar{\ell}}(\varepsilon)z_{\bar{\ell}}\| \quad (\text{C.6})$$

where the scalars  $z_\ell$  are the components of the unit-norm vector  $\mathbf{z}$ . Applying the triangle inequality to (C.6), we have

$$\begin{aligned} \|\mathbf{K}(\varepsilon)\mathbf{z}\| &= \|\mathbf{k}_1(\varepsilon)z_1 + \dots + \mathbf{k}_{\bar{\ell}}(\varepsilon)z_{\bar{\ell}}\| \leq \\ &\leq \|\mathbf{k}_1(\varepsilon)z_1\| + \dots + \|\mathbf{k}_{\bar{\ell}}(\varepsilon)z_{\bar{\ell}}\| = \\ &= \|\mathbf{k}_1(\varepsilon)\| |z_1| + \dots + \|\mathbf{k}_{\bar{\ell}}(\varepsilon)\| |z_{\bar{\ell}}| \leq \\ &\leq \sqrt{\bar{m}} (|z_1| + \dots + |z_{\bar{\ell}}|) \end{aligned} \quad (\text{C.7})$$

where the last inequality stems from (C.3). In the term  $(|z_1| + \dots + |z_{\bar{\ell}}|)$ , we recognize the 1-norm of vector  $\mathbf{z}$ . A well-known result is that  $\|\mathbf{z}\|_1 \leq \sqrt{\bar{\ell}} \|\mathbf{z}\|_2$  thus, recalling that  $\|\mathbf{z}\|_2 = 1$ , we have  $\|\mathbf{z}\|_1 \leq (\bar{\ell})^{1/2}$ . Hence, from (C.7) we have

$$\|\mathbf{K}(\varepsilon)\mathbf{z}\| \leq \sqrt{\bar{m}} (|z_1| + \dots + |z_{\bar{\ell}}|) \leq \sqrt{\bar{m}\bar{\ell}} \quad (\text{C.8})$$

All the above steps are valid independently of the choice of  $\varepsilon$  and  $\mathbf{z}$ , we can thus conclude that

$$\bar{\sigma}_{\mathbf{K}}(\varepsilon) \leq \sqrt{\bar{m}\bar{\ell}}, \quad \forall \varepsilon. \quad (\text{C.9})$$

## C.2 Proving the lower bound

Let us now prove that

$$\inf_{\varepsilon} \bar{\sigma}_{\mathbf{K}}(\varepsilon) \geq 1 \quad (\text{C.10})$$

Based on (C.5), the above is equivalent to

$$\inf_{\varepsilon} \sup_{\|\mathbf{z}\|_2=1} \|\mathbf{K}(\varepsilon)\mathbf{z}\| \geq 1. \quad (\text{C.11})$$

First, exploiting the min-max inequality [138] we have the that

$$\inf_{\varepsilon} \sup_{\|\mathbf{z}\|_2=1} \|\mathbf{K}(\varepsilon)\mathbf{z}\| \geq \sup_{\|\mathbf{z}\|_2=1} \inf_{\varepsilon} \|\mathbf{K}(\varepsilon)\mathbf{z}\| \quad (\text{C.12})$$

Focusing on the right-hand-side term, let us analyze the innermost minimization problem  $\inf_{\varepsilon} \|\mathbf{K}(\varepsilon)\mathbf{z}\|$ . By definition we have

$$\inf_{\varepsilon} \|\mathbf{K}(\varepsilon)\mathbf{z}\| \leq \inf_{\varepsilon} \|\mathbf{K}(\varepsilon)\| \|\mathbf{z}\| \quad (\text{C.13})$$

and, following (C.4), the minimum is attained at  $\varepsilon \rightarrow \infty$  where  $\|\mathbf{K}(\varepsilon)\| = 1$ . We conclude that,

$$\inf_{\varepsilon} \|\mathbf{K}(\varepsilon)\mathbf{z}\| \leq \|\mathbf{K}(\infty)\| \|\mathbf{z}\| = \|\mathbf{z}\| \quad (\text{C.14})$$

therefore

$$\sup_{\|\mathbf{z}\|_2=1} \inf_{\varepsilon} \|\mathbf{K}(\varepsilon)\mathbf{z}\| \leq \|\mathbf{z}\|_2 = 1 \quad (\text{C.15})$$

The above (C.15) holds with equality. This can be easily proven recalling that, for  $\varepsilon \rightarrow \infty$ , the columns of  $\mathbf{K}(\varepsilon)$  coincide with the canonical basis vectors  $\mathbf{e}_\ell$  (see Equation (6.3)). Taking  $\mathbf{z}^* = \mathbf{e}_\ell$ , for some  $\ell = 1, \dots, \bar{\ell}$  we have that  $\|\mathbf{K}(\infty)\mathbf{z}^*\| = 1$ . We can thus conclude that

$$\sup_{\|\mathbf{z}\|_2=1} \inf_{\varepsilon} \|\mathbf{K}(\varepsilon)\mathbf{z}\| = 1 \quad (\text{C.16})$$

Finally, plugging (C.16) in (C.12) leads to

$$\inf_{\varepsilon} \sup_{\|\mathbf{z}\|_2=1} \|\mathbf{K}(\varepsilon)\mathbf{z}\| \geq 1 \quad (\text{C.17})$$

that concludes our proof.

□