POLITECNICO DI TORINO

Master of Science in Mechatronic Engineering

Master Thesis

Adaptive algorithms for passivity verification and enforcement of multi-parametric behavioral macromodels

Supervisor
Stefano Grivet-Talocia

Candidate
Alessandro ZANCO
ID Number 231739

Academic Year 2017-2018

To mum, dad and Marta...

Acknowledgements

I would like to thank my supervisor, Prof. Stefano Grivet-Talocia, for his patience and enthusiasm in guiding me through this work, with precious suggestions and criticisms.
I would also like to thank my colleagues and friends, Marco and Tommaso, whose efforts have been essential to finalize this work.
Warm thanks go to my closest friends, whose presence encouraged me to do always my best.
I am profoundly grateful to my family, for supporting me in these years, letting me lay the foundations for my future.
Finally, I would like to dedicate this work to Marta, for always being alongside me and for believing in me, more than I have ever done.

Summary

Nowadays, designing complex interconnected electronic systems heavily relies on numerical simulations, in what is called Electronic Design Automation (EDA). With this acronym, a broad category of software tools aimed to support engineers in the development process is intended. These design techniques are even more important in optimization phases when focused what-if analyses are performed. Most often, these numerical simulations require massive computing resources and long runtimes. A well-established approach for speeding up these analyses is based upon macro-models. A macro-model is intended to be a behavioral reduced-order model that, after careful identification from a restricted set of known input/output responses, enables accurate simulations while keeping computational times manageable. A further boost to this method comes along with the parameterization of such models that, embedding an explicit dependence on design parameters, greatly simplifies the whole system optimization and design centering process. Two fundamental requirements that a parametric macro-model must fulfil to guarantee numerically reliable simulations are uniform stability and passivity throughout the parameter space. A non-passive model may lead to instability, causing a general failure in the system verification process. Unfortunately, standard model fitting algorithms are generally not able to ensure these requirements, and ad-hoc post-processing techniques must be used to impose these conditions for any parameter combination. This work, starting from state-of-the-art passivity verification techniques, proposes an effective and reliable extension to first check, and then enforce if necessary, the model passivity throughout a possibly multivariate parameter space. The main theoretical tool that is used in this approach is the so-called Hamiltonian matrix (or pencil) associated with the model, whose imaginary eigenvalues are known to provide a purely algebraic test for passivity. A single Hamiltonian eigenvalues computation is sufficient for non-parameterized models, but repeated eigenvalues extractions are necessary throughout the possibly high-dimensional parameters space in case of multivariate models. The main contribution of this work is the formulation and implementation of an adaptive method to sample the parameter space, looking for passivity violations. In contrast with previous approaches, this algorithm relies on first-order perturbations of Hamiltonian eigenvalues as a metric to drive the sampling algorithm, enabling checking for passivity violations only a minimum and strictly required number of parameters combinations. The major improvement of the proposed approach is its predictive property, which allows to outlook via a first-order approximation the passivity behavior of the system, even at regions in the parameters space that are not sampled. It results that, when using this adaptive sampling scheme within a passivity enforcement scheme, that iteratively perturbs model coefficients until

passivity is reached, major savings in CPU times were observed with respect to brute-force uniform fine-grid sampling approaches. The proposed algorithm was successfully tested on a large number of test-cases, some of which designed on purpose to stress the capability of the scheme to capture even very small passivity violations, and some other coming from real design parts obtained from industrial partners. We can safely state that the obtained results constitute the first available, reliable, multivariate passivity check and enforcement scheme. For this reason, a few conference papers have been submitted and an extended journal paper is in preparation.

Contents

List of Tables

List of Figures

Preface

The present thesis project is self-consistent and it has been developed independently by the Author, under the supervisor guidance. However, the results that have been achieved are of practical interest only if cast in a more general framework, in which other two thesis projects are involved: the shared effort of a team composed of Tommaso Bradde, Marco De Stefano and Alessandro Zanco enabled each member of the group to finalize his work. Being part of a joint effort, each of the three thesis projects shares a common background, which has been summarized in Chapters 1 and 2. These two chapters were written jointly and are common to all three thesis projects. The remaining chapters of each dissertation are the core of each project and are original for each individual team member.

Chapter 1

General Framework and Motivations

This chapter is co-authored by T. Bradde, M. De Stefano and A. Zanco.

1.1 Data Driven Modeling

This thesis project concerns mathematical modeling of linear dynamic systems, namely, systems that are governed by linear differential equations. By "mathematical modeling", we refer to the procedure by means of which a representation of a physical phenomenon or structure is given in a numerically (i.e quantitatively) exploitable form. This kind of representation grants us the opportunity to describe and predict what would happen in a given scenario in which the described object is involved; we can say that such a procedure is at the same time the foundation and the objective of science and a necessary step of the design process in every engineering field.
Although the first-principle laws of science are theoretically able to properly describe a broad range of dynamic phenomena, usually making use of partial differential equations, it is often inappropriate or impossible to derive from them a model able to satisfy the requirements of a current design process: the (exponentially) increasing complexity of the structures to be modeled would lead to an excessive computational cost with respect to the need of an easily manageable description of the item under design. Further, a model derived from first-principle laws must take care of all the physical quantities involved in the system dynamic, while often, only a subset of them is of practical interest.
The Data Driven Modeling techniques are intended to overcome these issues and to provide simpler yet accurate descriptions, able to catch the case-relevant aspects of the structures under investigation by exploiting, as common ground, a set of data collected to extract information about the system behaviour. Making use of proper algorithms, a suitable reconstruction of the relations underlying the data is achieved.
To gather the data, one can either carry out physical measurements of the desired quantities to be tracked or perform (once) a set of first-principle simulations from which the simplified model can be derived.
The most appropriate algorithm to process the data is always a matter of purposes, since

the structure of the algorithm influences, in some measure, the structure of the final model.
Beside the possible implementations, a broad spectrum classification of those algorithms can be based upon the a priori assumptions about the structure of the system: in the so called white and gray box approaches, a total or partial knowledge of the structure is assumed and the algorithm is expected to give back some quantities that characterize the imposed structure from the physical point of view; on the other hand, black box approaches make no assumptions on this structure and make no claims towards a physical description of the system, focusing only into the construction of models that fit numerically the data of the input-output relationship.
The first class of methods can give a deeper insight into the system behaviour, but they rely on the goodness of the a priori assumptions, that can result to be inaccurate or not possible to be made at all. Conversely, the lack of physical meaning of a black box model is counterbalanced by the opportunity to derive an input-output description without any assumption beyond linearity.
From now on, we will treat the black box methods and we will refer to the obtained model as "Macromodel".

1.1.1 Macromodels: construction flow and advantages

In the following, we will focus on macromodels devoted to the behavioural simulation of complex electrical interconnects, or, more generally, electromagnetic structures. The main objective of the macromodeling procedure is to obtain a macromodel that replaces the high complexity dynamics of the structure with a lower complexity model, which catches only the main features of the relationship between the electrical inputs and outputs of interest.
If we are modeling the system in the frequency domain, our starting point is a set of input-output data:

$$\breve{H}_k = \breve{H}(s_k) \quad for \quad k = 1,2,...,K$$

where s_k denotes a complex frequency and $\breve{H}(s_k)$ is the transfer function of the system sampled at s_k. The total number of measurements is K.
In most cases, the measurements are performed at the real frequencies $\mathrm{j}\omega_k$, with $s_k = \mathrm{j}\omega_k$. In this case we have:

$$\mathrm{j}\omega_1 = \mathrm{j}\omega_{min}, \quad \mathrm{j}\omega_K = \mathrm{j}\omega_{max}$$

The objective is then to reconstruct the response by means of an interpolation or approximation procedure that returns a model:

$$H(\mathrm{j}\omega) \approx \breve{H}(\mathrm{j}\omega) \quad for \quad \omega \in [\omega_{min}, \omega_{max}]$$

Throughout this text, we will denote with the symbol $\breve{H}(\cdot)$ the true system response, while with the symbol $H(\cdot)$ the model response. The obtained model is intended to be exploited in a circuit simulation software such as SPICE or EMTP in a fast and reliable way.
We now present a brief overview of how a macromodel is usually obtained and of the strong points that makes it useful.

- **Macromodeling from field solver data**: a full-wave solver is used to obtain the input-output data; detailed knowledge of the structures and of the characteristics of the actual system is required to perform the primary simulation. The data can be collected both in the time domain or in the frequency domain.
  This method is not properly a black-box one, since the structure of the model must be known to perform the full-wave simulation; anyway, we can say that it is a black box method for what concerns the macromodeling algorithm, that receives only data as inputs, without additional informations about structure. This scenario is common in industrial design environments where commercial field solvers are used.
- **Macromodeling from measurements**: a physical realization of the system under modeling is provided; the data are collected and reconstructed by performing measurements over the electrical ports that we wish to characterize. Also in this case, both frequency and time domain data can be gathered. This approach is truly black-box, in every step of the identification procedure.

Once the data are processed by the chosen algorithm, one can dispose of the obtained macromodel with the following advantages:

1. A closed form expression for the behaviour of the system is obtained from the discrete set of data points collected.
2. The macromodel describes the system behaviour without disclosing any insight about the physical structure: sharing a macromodel doesn't represent a risk for the diffusion of proprietary information.
3. Whatever is the nature of the data set used for the fitting, the resulting macromodel is intended to permit fast time domain simulations.
4. The obtained macromodel can be interfaced with other macromodels for simulation of large interconnects system, allowing the possibility to simulate and optimize various design scenarios.

1.1.2 Macromodel requirements for simulations

Some features are required on the macromodel, in order to guarantee its exploitability and reliability. In particular, since we are dealing with the modeling of linear systems, a suitable model structure should be chosen among all the possible ones; indeed, we know that when a system is governed by ordinary differential equations, all the transfer functions that can be derived for its input-output description result to be rational functions of the Laplace variable s.
The choice of a model structure of this type not only catches the underlying governing laws of the system, but results also particularly appropriate to be exploited to perform simulations driven by linear circuit simulation software.
The numerical precision of the model must always be consistent with some physical characteristics of the modeled structure to reproduce its behavior correctly; here, we list the most relevant in an intuitive fashion, leaving a more precise description to later sections.

- **Realness.** Although the rational macromodels make use of complex variables to describe the input-output behaviours, all the simulated quantities must be real numbers when observed in the time domain.

- **Causality.** Any physical system at rest can change its state only as a result of an external stimulus; for an input-output description, this fact implies the necessity of the output to be temporally preceded by its cause, the input.

- **Stability.** The concept of stability can be provided with various definitions; in the following we define stable a model whose poles show negative real part, that is, if $\{p_i\}$ is the set of poles of the model, then:

$$\mathrm{Re}\,\{p_i\} < 0 \quad for \quad i = 1,2,...,n$$

  where n is the order of the associated transfer function. The lack of the stability property can imply numerically unbounded simulations that clearly do not reflect the behavior of a real system.

- **Passivity.** A system is passive if it is not able to generate energy on its own; it can realease energy to the outer environment only if that energy was previously provided and stored inside it. The property of passivity can be regarded as the most general, since it automatically implies stability, causality and realness.

1.1.3 Rational fitting algorithms

The choice of a particular fitting strategy is the first step in any modeling procedure: we must first fix the structure of our model in order to restrict the set of all the possible candidate representations. Since our aim is to model electrical interconnects and their frequency-dependent behavior, the system will intrinsically exhibit a linear relationship between input and output, due to the nature of the electromagnetic phenomena.
It is well known that any linear system is fully characterized by a rational function of the complex variable s through its input-output transfer function:

$$H(s) = \frac{N(s)}{D(s)} \tag{1.1}$$

where $N(s)$ and $D(s)$ are polynomials.
Therefore, a natural choice is to try to reconstruct the system through a rational fitting procedure, that returns a model potentially able to catch all the information of interest.

Rational fitting algorithms make use of rational functions as basis for the model. Rational functions are universal approximators: any set of data can be fitted by a series of rational functions if a suitable order (i.e. number of basis functions) is considered. Even if this is for sure an encouraging starting point, several issues affect a modeling process relying on rational fitting:

- The behavior of the returned model is very accurate at the fitting points, but might show an unwanted and improper oscillating nature between the data points and beyond the limits of the data interval; this is particularly common when a very high

order for the interpolating function is chosen.
This phenomenon is known as over-fitting and must be taken into account during the identification procedure: one should use a subset of the available data to test the model quality at points of the domain that are not exploited for the fitting procedure.

- The imposition of constraints that ensure the model physical consistency can prevent the convergence of rational fitting algorithms or, most often, be the cause of a poor quality of the fitting.

From now on, we will assume that the model to be identified is a proper rational function of the variable s, although an extension to the improper case is straightforward.
The unknowns that the rational fitting algorithm is intended to return depend on the formulation of the rational function that we want to use. This formulation is fundamental because, as we will see, it can cast the model in forms that are more suitable with respect to others to achieve a good approximation. We now present the most common formulations of rational functions together with the unknowns that an algorithm is expected to return when such formulations are used as starting point.

- **Ratio of polynomials**: in this case, we assume that the model is representative of an underlying dynamics expressed as:

$$H(s;\boldsymbol{x}) = \frac{N(s;\boldsymbol{x})}{D(s;\boldsymbol{x})} = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_m s^m}{b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-1} s^{n-1} + s^n}$$

  in this case, the unknown vector $\boldsymbol{x}$ collects the $2n$ parameters:

$$\boldsymbol{x} = [a_0, a_1, a_2, \ldots, a_m, b_0, b_1, b_2, \ldots, b_{n-1}]^\mathsf{T}$$

  and the quality of the fitting can be evaluated by means of the residual quantity:

$$r_k(\boldsymbol{x}) = \breve{H}_k - \frac{a_0 + a_1 s_k + a_2 s_k^2 + \cdots + a_m s_k^m}{b_0 + b_1 s_k + b_2 s_k^2 + \cdots + b_{n-1} s_k^{n-1} + s_k^n}$$

  evaluated for each of the data samples.

- **Pole-zero form**: with this formulation the rational function reads:

$$\breve{H}(s,\boldsymbol{x}) = \alpha \frac{\prod_{j=1}^{n-1}(s - z_j)}{\prod_{j=1}^{n}(s - p_j)}$$

  the $2n$ unknown vector is now:

$$\boldsymbol{x} = [\alpha, z_1, z_2, \ldots, z_{n-1}, p_1, p_2, \ldots, p_n]^\mathsf{T}$$

  and each residual quantity is evaluated as:

$$r_k(\boldsymbol{x}) = \breve{H}_k - \alpha \frac{\prod_{j=1}^{n-1}(s - z_j)}{\prod_{j=1}^{n}(s - p_j)}$$

- **Partial fractions form**: in this case, the rational function is expressed as a series of partial functions of the form:

$$H(s, \boldsymbol{x}) = \sum_{j=1}^{n} \frac{c_j}{s - p_j} \tag{1.2}$$

with the assumption that the multiplicity of each pole equals one, that is:

$$p_i \neq p_j \quad \forall i \neq j.$$

The $2n$ unknown vector is now defined as:

$$\boldsymbol{x} = [c_1, c_2, \ldots, c_n, p_1, p_2, \ldots, p_n]^{\mathsf{T}}$$

and the residual quantities are:

$$r_k(\boldsymbol{x}) = \breve{H}_k - \sum_{j=1}^{n} \frac{c_j}{s - p_j}$$

- **Ratio of rational functions**: to formulate the model in this form, we observe first that any rational function of the variable s can be expressed as a ratio of other two rational functions in s; for this reason, we can cast the model in a more general form that reads:

$$H(s; \boldsymbol{x}) = \frac{N(s; \boldsymbol{x})}{D(s; \boldsymbol{x})} = \frac{\sum_{i=1}^{n} c_i \varphi_i(s)}{\sum_{i=1}^{n} d_i \varphi_i(s)} \tag{1.3}$$

where both numerator and denominator are expressed as a sum of *rational basis functions* $\varphi_i(s)$. In this case, the unknowns vector embeds the $2n$ coefficients of the series expansions:

$$\boldsymbol{x} = [c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n]^{\mathsf{T}}$$

while the residual vector is defined as:

$$r_k(\boldsymbol{x}) = \breve{H}_k - \frac{\sum_{i=1}^{n} c_i \varphi_i(s)}{\sum_{i=1}^{n} d_i \varphi_i(s)}$$

1.2 Rational fitting with fixed poles

Our main attempt is to formulate the rational fitting problem in such a way that a linear dependence holds between the unknowns and the basis functions that we want to use to fit the data. If this linear relation holds, then the rational fitting problem can be solved by means of a standard least squares problem: the basis functions are sampled in the points of the domain for which data points are available and the resulting numerical values are used to build the *regressor* matrix of the least square problem.
We can see how, among all the formulations of a rational function, the only one that can guarantee linearity between the unknowns and the basis functions is the partial fractions expansion (1.2) under the assumption that the poles p_j are fixed apriori. This formulation will be deeply exploited in the following since it allows the formulation of the rational fitting problem as a standard least squares problem.

1.2.1 Partial Fractions

We usually define the frequency-dependent basis functions, due to their very convenient numerical properties, as a set of partial fractions with a fixed set of poles. In particular, we realize a prescribed set of distinct $\bar{n}_r$ real poles $q_i \in \mathbb{R}^-$ and $\bar{n}_c$ complex pole pairs $q_{i,i+1} = q_i' \pm \mathrm{j}q_i'' \in \mathbb{C}^-$, where $\varphi_0(s) = 1$. The total number of basis functions is assumed to be $n = 1 + \bar{n}_r + 2\bar{n}_c$, including the constant term. We can define:

$$\text{if } \bar{q}_i \in \mathbb{R} \rightarrow \varphi_i(s) = (\bar{s} - \bar{q}_i)^{-1};$$

$$\text{if } \bar{q}_i \in \mathbb{C} \rightarrow \begin{cases} \varphi_i(s) = (\bar{s} - \bar{q}_i)^{-1} \\ \varphi_{i+1}(s) = \varphi_i^*(s) = (\bar{s} - \bar{q}_i^*)^{-1} \end{cases}$$

To improve numerical conditioning, this basis definition is based on normalized independent variables and poles, defined as:

$$\bar{s} = \frac{s}{\omega_0}, \quad \bar{q}_i = \frac{q_i}{\omega_0}$$

where ω_0 is a scaling frequency, which is in general obtained considering the largest model pole.

1.2.2 Least squares formulation of the fitting problem

Denoting with $\varphi_i(s)$ the generic element of our basis of partial fraction defined over a set of poles $\{q_i\}$, with $i = 0,1,2,...,n$, then the residual quantities related to each data sample can be written as:

$$r_k(\boldsymbol{x}) = \breve{H}_k - \boldsymbol{\varphi}_k^{\mathsf{T}} \boldsymbol{x}$$

with

$$\boldsymbol{\varphi}_k^{\mathsf{T}} = [\varphi_1(s_k), \varphi_2(s_k), ..., \varphi_n(s_k)], \quad \boldsymbol{x} = [c_1, c_2, ..., c_n]^{\mathsf{T}}$$

If we drop the dependency of the residuals on $\boldsymbol{x}$ we can write the above relationship in matrix form by writing:

$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_K \end{pmatrix} = \begin{pmatrix} \breve{H}_1 \\ \breve{H}_2 \\ \vdots \\ \breve{H}_K \end{pmatrix} - \begin{pmatrix} \boldsymbol{\varphi}_1^{\mathsf{T}} \\ \boldsymbol{\varphi}_2^{\mathsf{T}} \\ \vdots \\ \boldsymbol{\varphi}_K^{\mathsf{T}} \end{pmatrix} \boldsymbol{x}.$$

We can use the more compact and general notation:

$$\boldsymbol{b} = \begin{pmatrix} \breve{H}_1 \\ \breve{H}_2 \\ \vdots \\ \breve{H}_K \end{pmatrix}, \quad \boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_K \end{pmatrix}, \quad \boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\varphi}_1^{\mathsf{T}} \\ \boldsymbol{\varphi}_2^{\mathsf{T}} \\ \vdots \\ \boldsymbol{\varphi}_K^{\mathsf{T}} \end{pmatrix}$$

and write:

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{\Phi}\boldsymbol{x}$$

Since our goal is to minimize the value of the residuals, we can solve the least squares problem [26, 27]:

$$\boldsymbol{\Phi x} \approx \boldsymbol{b}$$

that returns an unknown vector $\boldsymbol{x}^*$ such that the euclidean norm of the vector $\boldsymbol{r}$ is minimized.
By writing the matrix $\boldsymbol{\Phi}$ in extended form we obtain the Cauchy matrix:

$$\boldsymbol{\Phi} = \begin{pmatrix} 1 & \frac{1}{s_1 - q_1} & \frac{1}{s_1 - q_2} & \dots & \frac{1}{s_1 - q_n} \\ 1 & \frac{1}{s_2 - q_1} & \frac{1}{s_2 - q_2} & \dots & \frac{1}{s_2 - q_n} \\ 1 & \vdots & \vdots & & \vdots \\ 1 & \frac{1}{s_K - q_1} & \frac{1}{s_K - q_2} & \dots & \frac{1}{s_K - q_n} \end{pmatrix},$$

It is well known that the condition number [16, 23] of the normal equations associated to the regressor matrix,

$$\kappa(\boldsymbol{\Phi}) = \sqrt{\frac{\sigma_{max}(\boldsymbol{\Phi}^H \boldsymbol{\Phi})}{\sigma_{min}(\boldsymbol{\Phi}^H \boldsymbol{\Phi})}}$$

strongly influences the quality of the solution of the least squares problem. Fortunately, being the partial fraction basis linearly independent (although not orthogonal) the Cauchy matrix is usually well conditioned.

1.3 General Rational Fitting

The situation explained in the previous section is desirable to solve the fitting problem, but it is very uncommon to known a priori the set of poles of the underlying system. For this reason, black box rational fitting algorithms must be able to return a model without any initial assumption beyond linearity. Two example of such algorithms are the Generalized Sanathanan-Koerner iteration (GSK), introduced in the following, and the Vector Fitting Iteration, for which a discussion can be found in [21].

1.3.1 Generalized Sanathanan Koerner iteration

The GSK [32]iteration makes use of the model formulation (1.3) to iteratively solve a linearized version of the rational fitting problem. At each iteration ν of the algorithm a modified residual quantity, defined as:

$$r_k^{\nu}(\boldsymbol{x_\nu}) = \frac{D(s_k; \boldsymbol{x}_\nu)\breve{H}_k - N(s_k; \boldsymbol{x}_\nu)}{D(s_k; \boldsymbol{x}_{\nu-1})}, \quad for \quad k = 1, 2, .., K$$

is minimized in LS sense. In this formulation $D(s_k; \boldsymbol{x}_\nu)$ is the denominator of the model at the current iteration (that is the one that will be found after the solution of the LS problem), while $D(s_k; \boldsymbol{x}_{\nu-1})$ is the denominator of the model computed at the previous iteration, evaluated at the fitting points. We denote with $\boldsymbol{x}_\nu$ an iteration-dependent

unknowns vector.
The idea behind the GSK algorithm is that as the number of iteration increases, the estimate of the denominator stabilizes, implying that the residual quantity becomes for $\nu \to \infty$:

$$r_k^{\nu}(\boldsymbol{x}_{\infty}) = \breve{H}_k - \frac{N(s_k;\boldsymbol{x}_{\infty})}{D(s_k;\boldsymbol{x}_{\infty})} \quad for \quad k = 1, 2, .., K$$

which coincides with the residual that we actually want to minimize. When the model is cast in the form (1.3), then the components of the residual vector $\boldsymbol{r}^{\nu}(\boldsymbol{x}_{\nu})$ at iteration ν will read:

$$r_k^{\nu}(\boldsymbol{x}_{\nu}) = \frac{[\varphi_0(s_k) + \sum_{j=1}^{n} d_j^{\nu}\varphi_j(s_k)]\breve{H}_k - \sum_{j=0}^{n} c_j^{\nu}\varphi_j(s_k)}{\varphi_0(s_k) + \sum_{j=1}^{n} d_j^{\nu-1}\varphi_j(s_k)}$$

Here we imposed $d_0 = 1$ to guarantee a unique solution of the system since the component φ_0 is usually associated with a constant term. We made all the coefficients iteration-dependent.
The iterative minimization of $||\boldsymbol{r}^{\nu}(\boldsymbol{x}_{\nu})||$ is achieved through the least square solution of the system:

$$(\mathbf{M}_{\nu-1}\boldsymbol{\Psi})\boldsymbol{x}_{\nu} \approx \mathbf{M}_{\nu-1}\boldsymbol{b}$$

where:

$$\mathbf{M}_{\nu-1} = diag\{m_1^{\nu-1}, m_2^{\nu-1}, ..., m_K^{\nu-1}\}, \quad m_k^{\nu-1} = \frac{1}{D(s_k;\boldsymbol{x}_{\nu-1})},$$

$$\boldsymbol{b} = [\breve{H}_1\varphi_0(s_1), \breve{H}_2\varphi_0(s_2), ..., \breve{H}_K\varphi_0(s_K)]^{\mathsf{T}},$$

$$\boldsymbol{\Psi} = [\boldsymbol{\Phi}_0,\ -\breve{\mathbf{H}}\boldsymbol{\Phi}_1],$$

$$\boldsymbol{\Phi}_0 = \begin{pmatrix} \varphi_0(s_1) & \varphi_1(s_1) & ... & \varphi_n(s_1) \\ \varphi_0(s_2) & \varphi_1(s_2) & ... & \varphi_n(s_2) \\ \vdots & \vdots & & \vdots \\ \varphi_0(s_K) & \varphi_1(s_K) & ... & \varphi_n(s_K) \end{pmatrix}$$

$$\boldsymbol{\Phi}_1 = \begin{pmatrix} \varphi_1(s_1) & \varphi_2(s_1) & ... & \varphi_n(s_1) \\ \varphi_1(s_2) & \varphi_2(s_2) & ... & \varphi_n(s_2) \\ \vdots & \vdots & & \vdots \\ \varphi_1(s_K) & \varphi_2(s_K) & ... & \varphi_n(s_K) \end{pmatrix}$$

The rational basis function that is usually exploited is the partial fractions basis.
We end this section by pointing out that the formulations of GSK we presented is given for the scalar case; anyway, a straightforward extension is possible to the multiport systems. For details see [21]. From now on, we will denote with the symbol $\breve{\mathbf{H}}(\cdot) \in \mathbb{C}^{P\times P}$ the multiport response of the true system and with $\mathbf{H}(\cdot) \in \mathbb{C}^{P\times P}$ the multiport response of our models, where the symbol P denotes the number of ports of the system. In the

following, we will describe the main model formulations used to characterize a multiport system macromodel.

1.4 Multiport (MIMO) Model Formulations

Approximating the true system response in a suitable macromodel form is fundamental to include the curve fitting result in system-level simulations using standard circuit solvers such as SPICE. Several mathematical structures are available: the identification algorithm efficiency, in frequency and time domain, is affected by this choice.
In this Section we are going to describe the model formulation through a transfer matrix and a state space realization; the latter will be useful for the macromodel characterization.

1.4.1 Transfer Function Formulation

Recalling to the scalar model of (1.1), we extend now the formulation realizing a rational model of a MIMO system. Considering a generic MIMO LTI system with rational transfer function, we can adopt the so-called *Generalized Sanathanan-Koerner (GSK)* form [33] [21] as:

$$\mathbf{H}(s) = \frac{\mathbf{N}(s)}{\mathsf{D}(s)} = \frac{\sum_{n=0}^{\bar{n}} \mathbf{R}_n \, \varphi_n(s)}{\sum_{n=0}^{\bar{n}} r_n \, \varphi_n(s)} \tag{1.4}$$

where we denoted with $\mathbf{R}_n \in \mathbb{R}^{P \times P}$ and $r_n \in \mathbb{R}$ the numerator and denominator model (real-valued) coefficients, respectively.
Frequency variations are induced by chosen basis function $\varphi_n(s)$, which are rational functions of s , with $\bar{n}$ frequency basis order.
Both numerator and denominator of (1.4) share the same basis poles set, which are assumed stable.

1.4.2 State Space and Descriptor Realizations

We now explore the state space and descriptor realizations of a MIMO LTI system, starting from the pole-residue or GSK form of the model $\mathbf{H}(s)$ in the Laplace domain.

State Space for Pole-Residue Form

Considering a general P-ports model in a pole-residue form, we can write:

$$\mathbf{H}(s) = \mathbf{H}_\infty + \sum_{n=1}^{\bar{n}} \frac{\mathbf{R}_n}{s - q_n} \tag{1.5}$$

where $\mathbf{H}_\infty = \mathbf{R}_0$ and $\bar{n}$ is the overall number of poles.
Denoting as:

$$\begin{aligned}\mathbf{A} &= \text{blkdiag}\{q_n \mathbb{I}_P\}_{n=1}^{\bar{n}_r} \\ \mathbf{B} &= \big[1,\ldots,1\big]^{\mathsf{T}} \otimes \mathbb{I}_P \\ \mathbf{C} &= \big[\mathbf{R}_1,\ldots,\mathbf{R}_{\bar{n}}\big] \\ \mathbf{D} &= \mathbf{H}_\infty\,.\end{aligned}$$

with $\otimes$ the matrix Kronecker product, we define a regular state-space system realization as:

$$\begin{cases}\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t) + \mathbf{D}\boldsymbol{u}(t)\end{cases} \tag{1.6}$$

where $\boldsymbol{u}, \boldsymbol{y} \in \mathbb{R}^P$ are the system input and output, respectively, and $\boldsymbol{x} \in \mathbb{R}^N$ are the system internal states.
The notation that we provide for the state-space realization is the following:

$$\mathbf{H}(s) = \mathbf{D} + \mathbf{C}(s\mathbb{I} - \mathbf{A})^{-1}\mathbf{B} \leftrightarrow \left(\begin{array}{c|c}\mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D}\end{array}\right)$$

Considering now the model in (1.4), with $\varphi(s)$ defined as the partial-fraction basis with a prescribed set of poles q_n (see Section 1.2.1), we can write:

$$\begin{aligned}\mathbf{N}(s) &= \mathbf{R}_0 + \sum_{n=1}^{\bar{n}} \frac{\mathbf{R}_n}{s - q_n} \\ \mathsf{D}(s) &= r_0 + \sum_{n=1}^{\bar{n}} \frac{r_n}{s - q_n}\end{aligned}$$

We now construct the two separate state-space realizations for numerator and denominator as:

$$\begin{aligned}\mathbf{N}(s) &\leftrightarrow \left(\begin{array}{c|c}\mathbf{A}_0 & \mathbf{B}_0 \\ \hline \mathbf{C}_1(s) & \mathbf{D}_1(s)\end{array}\right) \\ \mathsf{D}(s)\mathbb{I}_P &\leftrightarrow \left(\begin{array}{c|c}\mathbf{A}_0 & \mathbf{B}_0 \\ \hline \mathbf{C}_2(s) & \mathbf{D}_2(s)\end{array}\right)\end{aligned} \tag{1.7}$$

where:

$$\begin{aligned}\mathbf{A}_0 &= \text{blkdiag}\{\mathbf{A}_{0r},\ \mathbf{A}_{0c}\} \\ \mathbf{B}_0^{\mathsf{T}} &= \big[\mathbf{B}_{0r}^{\mathsf{T}},\ \mathbf{B}_{0c}^{\mathsf{T}}\big] \\ \mathbf{C}_1 &= \big[\mathbf{R}_1,\ldots,\mathbf{R}_{\bar{n}}\big] \\ \mathbf{C}_2 &= \big[\mathbb{I}_P\, r_1,\ldots,\mathbb{I}_P\, r_{\bar{n}}\big] \\ \mathbf{D}_1 &= \mathbf{R}_0 \\ \mathbf{D}_2 &= \mathbb{I}_P\, r_0.\end{aligned}$$

with:

$$\begin{aligned}
\mathbf{A}_{0r} =& \text{blkdiag}\{q_n \mathbb{I}_P\}_{n=1}^{\bar{n}_r} \\
\mathbf{A}_{0c} =& \text{blkdiag}\left\{ \begin{bmatrix} q_n' \mathbb{I}_P & q_n'' \mathbb{I}_P \\ -q_n'' \mathbb{I}_P & q_n' \mathbb{I}_P \end{bmatrix} \right\}_{n=1}^{\bar{n}_c} \\
\mathbf{B}_{0r} =& \left[1, \ldots, 1\right]^\mathsf{T} \otimes \mathbb{I}_P \\
\mathbf{B}_{0c} =& \left[2, 0, \ldots, 2, 0\right]^\mathsf{T} \otimes \mathbb{I}_P
\end{aligned}$$

where real-valued matrices have been used for complex conjugate poles.
Cascading the expressions in (1.7), we obtain a compact state-space realization for the rational model $\mathbf{H}(s)$, that reads:

$$\mathbf{H}(s) = \mathbf{N}(s)(\mathsf{D}(s)^{-1} \mathbb{I}_P) \leftrightarrow \left(\begin{array}{c|c} \mathbf{A}_0 - \mathbf{B}_0 \mathbf{D}_2^{-1} \mathbf{C}_2 & \mathbf{B}_0 \mathbf{D}_2^{-1} \\ \hline \mathbf{C}_1 - \mathbf{D}_1 \mathbf{D}_2^{-1} \mathbf{C}_2 & \mathbf{D}_1 \mathbf{D}_2^{-1} \end{array} \right)$$

We recall [21] and [25] for more details.

Descriptor Form

We now define an alternative system realization, denoted as descriptor form (or differential-algebraic system of equations (DAE), see [21]) of an impulse free model as:

$$\begin{cases} \mathbf{E}\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t) \end{cases} \tag{1.8}$$

where $\boldsymbol{u}$ and $\boldsymbol{y}$ are the system input and output, respectively, and $\boldsymbol{x} \in \mathbb{R}^{N+P}$, with $N = \bar{n}P$, the system internal states: the number of states changes with respect to the state-space realization, increasing the problem dimension.
The descriptor matrices of (1.8) are:

$$\mathbf{E} = \begin{pmatrix} \mathbb{I}_N & \mathbf{0}_{N,P} \\ \mathbf{0}_{P,N} & \mathbf{0}_{P,P} \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{B}_0 \\ \mathbf{C}_2 & \mathbf{D}_2 \end{pmatrix}$$
$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_1 & \mathbf{D}_1 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} \mathbf{0}_{N,P} \\ -\mathbb{I}_P \end{pmatrix}$$

with $\mathbf{0}_{J,K}$ the null matrix of size $J \times K$.
It can be proven that the model expression of (1.4) is equivalent to

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

as detailed in [21].
The descriptor form is particularly useful because it requires no block matrix inversion and moreover all matrix elements depend linearly on the model coefficients, in opposition with the regular state space realization.

In the following sections we are going to describe in more details how the model should reflect the physical properties of the true system.

1.5 Stability

Several stability definitions may be formulated for an LTI system, analysing the general properties of all the possible solutions of a system. During our work we only modelled black-box systems, which can be characterized, from a stability standpoint, through the matrix $\mathbf{A}$ of the state-space realization (1.6) of the model $\mathbf{H}(s)$.
For this reason, we can define an LTI system [24] [30] [45] as:

- *asymptotically stable* if and only if all the poles have a strictly negative real part, $\mathrm{Re}\{q_n\} < 0 \; \forall n$;
- *stable* if and only if all the poles have a negative real part, $\mathrm{Re}\{q_n\} \leq 0 \; \forall n$, and all the purely imaginary poles have a multiplicity that is at most one;
- *unstable* if at least one pole has either a strictly positive real part, $\mathrm{Re}\{q_n\} > 0$, or a null real part with a multiplicity higher than one.

Furthermore, since the eigenvalues of $\mathbf{A}$ are the model poles q_n, the matrix $\mathbf{A}$ can be denoted as (*asymptotically*) *stable* if its eigenvalues have a (strictly) negative real part.

1.6 Passivity

In electronic systems engineering, it is a common practice to deal with many interconnected sub-systems. Especially during high-speed electronic devices design, it is fundamental to assess the signal and power integrity (SI, PI), when all the sub-systems are connected together, since even individual components like vias and packages may strongly affect SI and PI performances if the design is poor. In general, it is common to perform in-depth analyses of these components and, to speed-up the whole process, surrogate macro-models for each sub-system are used, that will be connected together just in simulation phases. Such analyses of interconnected systems may suffer from instabilities, even if all the models are internally asymptotically stable. In fact, if one or more of the single macro-models is not passive, an un-physical energy generation may occur, leading to a distorted output signal which may have detrimental effects on the whole system. This fact, under suitable load conditions, may be responsible of an uncontrolled amplification of the output signal, resulting in an unstable simulation.
Model passivity turns out to be a fundamental requirement that must be carefully analysed when such macro-models are synthesized to ensure reliable simulations under any working condition.

The passivity of a system is strongly related to the net power it absorbs at any time instant t. Considering a P-ports system, the absorbed instantaneous power is:

$$p(t) = \sum_{k=1}^{P} p_k(t) = \sum_{k=1}^{P} v_k(t) i_k(t)$$

that can be written in compact form as:

$$p(t) = \mathbf{v}(t)^{\mathsf{T}} \mathbf{i}(t) = \mathbf{i}(t)^{\mathsf{T}} \mathbf{v}(t)$$

where $\mathbf{v} = [v_1, \ldots, v_k]^\mathsf{T}$ and $\mathbf{i} = [i_1, \ldots, i_k]^\mathsf{T}$.

In case the system is in immittance representation, input and output variables, denoted respectively as $u_k(t)$ and $y_k(t)$ may be either voltage or currents. The instantaneous power is thus:

$$p(t) = \boldsymbol{y}^\mathsf{T}(t)\boldsymbol{u}(t) = \boldsymbol{u}^\mathsf{T}(t)\boldsymbol{y}(t)$$

Considering input and output as complex valued signals, the instantaneous power definition can be generalized, as:

$$p(t) = \mathrm{Re}\left\{\mathbf{v}^\mathsf{H}(t)\mathbf{i}(t)\right\} = \mathrm{Re}\left\{\mathbf{i}^\mathsf{H}(t)\mathbf{v}(t)\right\} \tag{1.9}$$

For scattering representations, voltages v_k and currents i_k are transformed in incident and reflected scattering waves, denoted respectively as a_k and b_k. To this end we recall that:

$$a_k = \frac{1}{2\sqrt{R_{ref,k}}}(v_k + R_{ref,k}i_k)$$
$$b_k = \frac{1}{2\sqrt{R_{ref,k}}}(v_k - R_{ref,k}i_k)$$

where $R_{ref} > 0$ is the normalization resistance of each port.
The power $p(t)$ for scattering representation is thus:

$$p(t) = \sum_{k=1}^{P} \sqrt{R_{ref,k}}[a_k(t) + b_k(t)]\frac{1}{\sqrt{R_{ref,k}}}[a_k(t) - b_k(t)] = \boldsymbol{a}(t)^\mathsf{T}\boldsymbol{a}(t) - \boldsymbol{b}(t)^\mathsf{T}\boldsymbol{b}(t)$$

with $\boldsymbol{a}(t) = [a_1(t), \ldots, a_k(t)]$ and $\boldsymbol{b}(t) = [b_1(t), \ldots, b_k(t)]$.
Defining generic input and output signals as $\boldsymbol{u}(t) = \boldsymbol{a}(t)$ and $\boldsymbol{y}(t) = \boldsymbol{b}(t)$, it follows that:

$$p(t) = \boldsymbol{u}(t)^\mathsf{T}\boldsymbol{u}(t) - \boldsymbol{y}(t)^\mathsf{T}\boldsymbol{y}(t)$$

Generalizing this definition to the case in which $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ are complex-valued signals, the instantaneous power is:

$$p(t) = \boldsymbol{u}(t)^\mathsf{H}\boldsymbol{u}(t) - \boldsymbol{y}(t)^\mathsf{H}\boldsymbol{y}(t) \tag{1.10}$$

The net energy absorbed by a P-ports system in a time interval $[t_1,\ t_2]$ is defined as:

$$\mathcal{E}(t) = \int_{t_1}^{t_2} p(\tau)d\tau \tag{1.11}$$

If the energy for $t_1 \to -\infty$ is vanishing, the cumulative net energy at an arbitrary time instant t is:

$$\mathcal{E}(t) = \int_{-\infty}^{t} p(\tau)d\tau \tag{1.12}$$

The definition for passivity now can be stated.

Definition 1.1 *[21, 45, 46] A P-ports system is passive if the cumulative net energy in* (1.12) *is non-negative for any time t*

$$\mathcal{E}(t) \geq 0, \ \forall t \tag{1.13}$$

and for any input signal $\boldsymbol{u}(t)$.

The term "passivity" is often replaced by its synonym "dissipativity", so that a passive system is also denoted as "dissipative".

1.6.1 The dissipation inequality

The passivity definition given in the previous section regards only the net input/output energy flow, without making any reference to the system internal energy. An alternative way to describe the passivity of a system is to relate the amount of energy it stores and exchanges with the environment, for any time t. Considering a generic system (described in its state space representation) the following dissipativity definition holds:

Definition 1.2 *[21] A system (expressed in its state space representation) is dissipative with respect to the supply rate* $p(t)$ *if there exist a scalar-valued function* $V(\boldsymbol{x})$*, with* $\boldsymbol{x}$ *the system states, such that*

$$V(\boldsymbol{x}(t_1)) \leq V(\boldsymbol{x}(t_0)) + \int_{t_0}^{t_1} p(t)dt, \ \forall \ t_0 \leq t_1 \, and \, \forall \ \boldsymbol{u}, \boldsymbol{y}, \boldsymbol{x}. \tag{1.14}$$

.

The integral term in (1.14) is exactly the net cumulative energy entering the system in the time interval $[t_0, \ t_1]$, as defined in (1.11). The function $V(\boldsymbol{x}(t))$ is recognized to be the energy that is stored by the system at any time instant t. Equation (1.14) states that in a system, to be dissipative, the variation on internal energy $V(\boldsymbol{x}(t_1)) - V(\boldsymbol{x}(t_0))$ can not exceed the energy that is supplied from the environment to the system during the time interval $[t_0, \ t_1]$.

If the storage function is differentiable, Equation (1.14) can be rewritten in differential form as:

$$\frac{d}{dt}V(\boldsymbol{x}(t)) \leq p(t), \ \forall t \tag{1.15}$$

Under the assumption that the energy stored for $t \to -\infty$ is vanishing, inequality (1.14) reduces to the passivity condition in Equation (1.13). This way to characterize the passivity of a system will turn out to be useful later on, when advanced algebraic passivity assessment methods will be derived

1.6.2 Passivity characterization

Considering now the class of MIMO (Multi Input-Multi Output) lumped LTI systems with input $\boldsymbol{u}(t)$ and output $\boldsymbol{y}(t)$, for which there exist a transfer matrix representation,

the previous dissipativity definition can be written in terms of the transfer function $\mathbf{H}(s)$, for both immittance and scattering representations.

For an immittance system, in order to derive passivity conditions in terms of its transfer matrix $\mathbf{H}(s)$, we can explicitly write the instantaneous absorbed power under cisoidal excitation $\boldsymbol{u}(s) = \boldsymbol{u}\ e^{st}$ using (1.9) as:

$$p(t) = \mathrm{Re}\left\{\boldsymbol{u}^{\mathsf{H}}\mathbf{H}\boldsymbol{u}\right\} e^{2\sigma t},\ \sigma = \mathrm{Re}\left\{s\right\} \tag{1.16}$$

The cumulative net energy can be computed as:

$$\mathcal{E}(t) = \int_{-\infty}^{t} p(\tau)d\tau = \mathrm{Re}\left\{\boldsymbol{u}^{\mathsf{H}}\mathbf{H}(s)\boldsymbol{u}\right\} \frac{e^{2\sigma t}}{2\sigma}$$

where $\sigma > 0$ to ensure the integral convergence.
Recalling the passivity condition in (1.13), it must hold $\mathcal{E}(t) \geq 0,\ \forall t$. Thus, being $\frac{e^{2\sigma t}}{2\sigma} > 0$ by assumption, it follows that:

$$\mathrm{Re}\left\{\boldsymbol{u}^{\mathsf{H}}\mathbf{H}(s)\boldsymbol{u}\right\} = \boldsymbol{u}^{\mathsf{H}}\left[\frac{1}{2}(\mathbf{H}(s) + \mathbf{H}^{\mathsf{H}}(s))\right]\boldsymbol{u} \geq 0,\ \forall \boldsymbol{u} \in \mathbb{C}^{P}$$

We can conclude that an immittance system is dissipative if:

$$\mathbf{H}(s) + \mathbf{H}^{\mathsf{H}}(s) \geq 0,\ \mathrm{Re}\left\{s\right\} > 0 \tag{1.17}$$

For further details on these derivations see [21].

To derive passivity conditions for scattering systems, as for the immittance case, we must write the instantaneous power in terms of $\mathbf{H}(s)$. Under a cisoidal excitation $\boldsymbol{u}(t)$, recalling Equation (1.10), the power $p(t)$ reads:

$$p(t) = \boldsymbol{u}(t)^{\mathsf{H}}\boldsymbol{u}(t) - \boldsymbol{y}(t)^{\mathsf{H}}\boldsymbol{y}(t) = \boldsymbol{u}^{\mathsf{H}}[\mathbb{I} - \mathbf{H}(s)^{\mathsf{H}}\mathbf{H}(s)]\boldsymbol{u}\ e^{2\sigma t}.$$

As for the immittance case, we compute the cumulative net energy absorbed by the system at time instant t as:

$$\mathcal{E}(t) = \int_{-\infty}^{t} p(\tau)d\tau = \boldsymbol{u}^{\mathsf{H}}\left[\mathbb{I} - \mathbf{H}^{\mathsf{H}}(s)\mathbf{H}(s)\right]\boldsymbol{u}\frac{e^{2\sigma t}}{2\sigma}$$

with $\sigma > 0$.
The passivity condition in (1.13) implies that:

$$\mathbb{I} - \mathbf{H}(s)^{\mathsf{H}}\mathbf{H}(s) \geq 0,\ \mathrm{Re}\left\{s\right\} > 0. \tag{1.18}$$

The two passivity conditions for immittance and scattering representation given above are now generalized with reference to Positive Real and Bounded Real matrices [2,21,44].

Definition 1.3 *A transfer matrix* $\mathbf{H}(s)$ *is Positive Real if:*

1. *each element of* $\mathbf{H}(s)$ *is defined and analytic in* $\mathrm{Re}\{s\} > 0$
2. $\mathbf{H}^*(s) = \mathbf{H}(s^*)$
3. $\mathbf{\Theta}(s) = \mathbf{H}(s) + \mathbf{H}^{\mathsf{H}}(s) \geq 0$ *for* $\mathrm{Re}\{s\} > 0$

where $\mathbf{H}^*(s)$ denotes the complex conjugate of $\mathbf{H}(s)$.

Definition 1.4 *A transfer matrix* $\mathbf{H}(s)$ *is Bounded Real if:*

1. *each element of* $\mathbf{H}(s)$ *is defined and analytic in* $\mathrm{Re}\{s\} > 0$
2. $\mathbf{H}^*(s) = \mathbf{H}(s^*)$
3. $\mathbf{\Theta}(s) = \mathbb{I} - \mathbf{H}^{\mathsf{H}}(s)\mathbf{H}(s) \geq 0$ *for* $\mathrm{Re}\{s\} > 0$

Condition 1 is related to stability and causality. In fact both causality and stability requires the transfer function to be analytic (must not have poles) in the closed right half complex plane.

Condition 2 may be interpreted as a "consistency" one, since it implies that the transfer matrix is real for any $s \in \mathbb{R}$. This condition strongly affects the residues of $\mathbf{H}(s)$: in fact, to be satisfied, they must be real, for real poles, or must appear in complex conjugate pairs, when corresponding to complex conjugate poles.

Finally, Condition 3 is exactly the one we derived above in Equations (1.17) and (1.18), related to the energy of the system described by $\mathbf{H}(s)$.
We are now ready to re-formulate LTI system passivity conditions in terms of Positive Real and Bounded Real matrices, as stated in Theorem 1.1 ([2, 21, 44]).

Theorem 1.1 *A LTI system with transfer matrix* $\mathbf{H}(s)$ *is defined to be passive if and only if* $\mathbf{H}(s)$ *is Positive Real (for immittance representations) or Bounded Real (for scattering representations).*

This theorem 1.1 provides a powerful theoretical tool to check the passivity of an LTI system through its transfer matrix. However, verifying that the three conditions are concurrently fulfilled in the open complex plane, implies considerable computational efforts. In the following, we derive some simpler conditions, based on the rational nature of the model underlying the transfer matrix $\mathbf{H}(s)$ to assess whether the model is passive, for both immittance and scattering representations.

Considering immittance systems, the following Theorem holds [2, 21, 44].

Theorem 1.2 *A rational matrix* $\mathbf{H}(s)$ *is Positive Real if and only if*

1. $\mathbf{H}(s)$ *has no poles in* $\mathbb{C}_+$
2. $\mathbf{H}^*(\mathrm{j}\omega) = \mathbf{H}(-\mathrm{j}\omega)$
3. $\mathbf{H}(\mathrm{j}\omega) + \mathbf{H}^{\mathsf{H}}(\mathrm{j}\omega) \geq 0,\ \forall\omega \in \mathbb{C}$*, except for simple poles* $\mathrm{j}\omega_i$ *of* $\mathbf{H}(s)$ *where the transfer matrix must be Hermitian and nonnegative definite.*
4. *for* $\omega \to \infty$, $\mathbf{H}(s) \sim \mathbf{R}_\infty s$ *in* $\mathrm{Re}\{s\} > 0$, *with* $\mathbf{R}_\infty$ *real, symmetric and non-negative definite*

The main advantage of this theorem with respect to the more general one, as shown in [21], is evident from the third condition. In fact, comparing it with the one defined in Definition 1.3, it turns out that the non-negative definiteness of $\mathbf{H}(s) + \mathbf{H}^{\mathsf{H}}(s)$ can be checked just along the imaginary axis rather than in the right half open complex plane. If Conditions 1,2,4 are satisfied (as usually are), the only thing we need to check is Condition 3, whose statement can be cast as follows:

$$\lambda_{min}(\mathrm{j}\omega) \geq 0,\ \forall\omega \in \mathbb{R}$$

with:

$$\lambda_{min}(\mathrm{j}\omega) = \min\{\lambda(\mathbf{H}(\mathrm{j}\omega) + \mathbf{H}^{\mathsf{H}}(\mathrm{j}\omega)\},\ \forall\omega \in \mathbb{R}$$

Assuming the transfer matrix to be asymptotically stable, the above eigenvalues are continuous functions of frequency, making in turn $\lambda_{min}(\mathrm{j}\omega)$ to be a continuous function of frequency.
This fact enables the use of frequency sampling techniques in advanced passivity assessment algorithms.
Now we are going to detail, for immittance and scattering systems, a set of fundamental results that enable the construction of advanced passivity verification techniques.

Immittance systems

Particularizing the dissipation inequality (1.15) for immittance LTI systems, we will derive a condition to assess system passivity in terms of the state-space representation matrices. To this end, we need to have an analytic expression of the supplied power that is given by Equation (1.16) and reads:

$$p(t) = \frac{1}{2}[\boldsymbol{u}^{\mathsf{T}}(\mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u}) + (\mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u})^{\mathsf{T}}\boldsymbol{u}]$$

If the storage function is defined as $V(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}^{\mathsf{T}}\mathbf{P}\boldsymbol{x})$, with $\mathbf{P}$ a symmetric positive definite matrix, its derivative (rate of change of the internal energy) will be:

$$\frac{d}{dt}V(\boldsymbol{x}(t)) = \frac{1}{2}[(\mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u})^{\mathsf{T}}\mathbf{P}\boldsymbol{x} + \boldsymbol{x}^{\mathsf{T}}\mathbf{P}(\mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u})]$$

Let us now impose the dissipativity condition defined in Equation (1.13). Splitting input and state signals, with trivial algebraic manipulations we get to the following LMI form, known as *Positive Real Lemma* [2, 34].

Lemma 1.1 *A LTI system in immittance form is passive if and only if, for any signal* $\boldsymbol{x}, \boldsymbol{u}$ *satisfying the state equations, it holds that:*

$$\exists \mathbf{P} = \mathbf{P}^{\mathsf{T}} > 0 : \quad \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} \mathbf{A}^{\mathsf{T}}\mathbf{P} + \mathbf{P}\mathbf{A} & \mathbf{P}^{\mathsf{T}}\mathbf{B} - \mathbf{C}^{\mathsf{T}} \\ \mathbf{B}^{\mathsf{T}}\mathbf{P} - \mathbf{C} & -(\mathbf{D} + \mathbf{D}^{\mathsf{T}}) \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix} \leq 0$$

We now derive a fundamental result (for details see [21]) used extensively in LTI passivity assessment algorithms, that enables the use of algebraic methods to spot passivity violations. In details, it will be shown that the imaginary eigenvalues of a particular Hamiltonian-structured matrix are strongly related to the location of passivity violations along the frequency axis.
First, let us define a support matrix function, called *Popov function*, $\boldsymbol{\Psi}(s)$ as:

$$\boldsymbol{\Psi}(s) = \mathbf{H}(s) + \mathbf{H}^{\mathsf{T}}(-s)$$

Recalling that, to be passive, the transfer matrix of an immittance system must satisfy:

$$\boldsymbol{\Theta}(s) = \mathbf{H}(s) + \mathbf{H}^{\mathsf{H}}(s) \geq 0$$

in turns out that $\boldsymbol{\Theta}(s)$ and $\boldsymbol{\Psi}(s)$ are equal when evaluated on the imaginary axis. This enables us to check the non-negative definiteness of $\boldsymbol{\Psi}(\mathrm{j}\omega)$ instead of $\boldsymbol{\Theta}(\mathrm{j}\omega)$.
The condition that must be verified to guarantee passivity is thus:

$$\boldsymbol{\Psi}(\mathrm{j}\omega) \geq 0, \forall \omega$$

Focusing our attention to this last equation, we see that the frequencies $\mathrm{j}\omega_i$ at which $\boldsymbol{\Psi}(\mathrm{j}\omega_i)$ becomes singular, algebraically pinpoint passivity violations, being $\boldsymbol{\Psi}(\mathrm{j}\omega_i)$ singular exactly when $\mathbf{H}(\mathrm{j}\omega_i) + \mathbf{H}^{\mathsf{T}}(-\mathrm{j}\omega_i) = 0$.
These frequencies are defined to be the solutions of:

$$\boldsymbol{\Psi}(\mathrm{j}\omega_i)\boldsymbol{u} = 0 \tag{1.19}$$

for some vector $\boldsymbol{u}$.
In order to algebraically find these solutions, we derive a state-space realization of $\boldsymbol{\Psi}(s)$, the analytic extension to the open complex plane of $\boldsymbol{\Psi}(\mathrm{j}\omega)$. This turns out to be useful since the solutions of Equation (1.19) are the poles of $\boldsymbol{\Psi}^{-1}(s)$, for which a simple state space realization is readily computed. The poles of $\boldsymbol{\Psi}^{-1}(s)$ are the eigenvalues of its dynamic matrix, that reads:

$$\boldsymbol{\mathcal{N}}_0 = \mathbf{A}_{\boldsymbol{\Psi}^{-1}} = \mathbf{A}_\Psi - \mathbf{B}_\Psi \mathbf{D}_\Psi^{-1} \mathbf{C}_\Psi$$

where $\mathbf{A}_\Psi,\ \mathbf{B}_\Psi,\ \mathbf{C}_\Psi,\ \mathbf{D}_\Psi$ are the state-space realization matrices of $\boldsymbol{\Psi}(s)$.
Expanding $\boldsymbol{\mathcal{N}}_0$ in terms of the system realization matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ we get the following matrix:

$$\boldsymbol{\mathcal{N}}_0 = \begin{pmatrix} \mathbf{A} - \mathbf{B}(\mathbf{D} + \mathbf{D}^{\mathsf{T}})^{-1}\mathbf{C} & -\mathbf{B}(\mathbf{D} + \mathbf{D}^{\mathsf{T}})^{-1}\mathbf{B}^{\mathsf{T}} \\ \mathbf{C}^{\mathsf{T}}(\mathbf{D} + \mathbf{D}^{\mathsf{T}})^{-1}\mathbf{C} & -\mathbf{A}^{\mathsf{T}} + \mathbf{C}^{\mathsf{T}}(\mathbf{D} + \mathbf{D}^{\mathsf{T}})^{-1}\mathbf{B}^{\mathsf{T}} \end{pmatrix}$$

Defining as $\mathbf{J}$ the following matrix:

$$\mathbf{J} = \begin{pmatrix} 0 & \mathbb{I}_n \\ -\mathbb{I}_n & 0 \end{pmatrix}$$

it holds that:

$$(\mathbf{J}\boldsymbol{\mathcal{N}}_0)^{\mathsf{T}} = \mathbf{J}\boldsymbol{\mathcal{N}}_0 \tag{1.20}$$

which shows that $\boldsymbol{\mathcal{N}}_0$ has a Hamiltonian structure.
Because of that, $\boldsymbol{\mathcal{N}}_0$ has some peculiar characteristics. In particular, its eigenspectrum is symmetric with respect to both imaginary and real axes. In fact the set of poles of $\boldsymbol{\Psi}(s)$ includes the ones of $\mathbf{H}(s)$ which are symmetric with respect to the real axis, and their mirror images, symmetric with respect to the imaginary axis.
The following theorem, proposed in [4,17,21], provides a fundamental results that relates the eigenvalues of $\boldsymbol{\mathcal{N}}_0$ with the ones of $\boldsymbol{\Psi}(\mathrm{j}\omega)$.

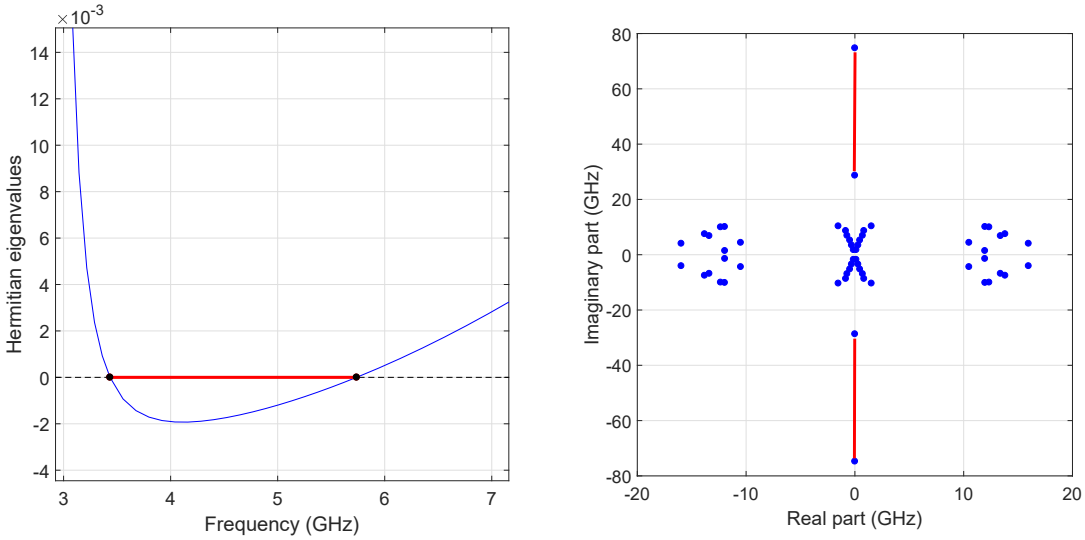
Theorem 1.3 *Let $\mathbf{H}(s)$ be the transfer matrix of an immittance system, whose state space matrices are $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, where $\mathbf{A}$ has no purely imaginary eigenvalues and $\mathbf{D} + \mathbf{D}^{\mathsf{T}}$ is non-singular. Then, $\mathrm{j}\omega_0$ is an eigenvalue of $\boldsymbol{\mathcal{N}}_0$ if and only if 0 is an eigenvalue of $\boldsymbol{\Psi}(\mathrm{j}\omega_0)$.*

It follows that, if $\boldsymbol{\mathcal{N}}_0$ has imaginary eigenvalues, the related LTI system is not passive for some frequency bands.
This result is formally stated in Theorem 1.4.

Theorem 1.4 *Let $\mathbf{H}(s)$ be the transfer matrix of an immittance system, whose state space matrices are $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, where $\mathbf{A}$ has no purely imaginary eigenvalues and $\mathbf{D} + \mathbf{D}^{\mathsf{T}}$ is positive definite. Then the system is passive if the Hamiltonian matrix $\boldsymbol{\mathcal{N}}_0$ has no purely imaginary eigenvalues.*

Theorems 1.3 and 1.4 provide an algebraic tool that is able to precisely verify system passivity and enables us to easily localize violation areas along the frequency axis.
To this end we must notice that Hamiltonian imaginary eigenvalues correspond to the complex frequencies at which at least one eigenvalue of $\boldsymbol{\Psi}(\mathrm{j}\omega)$ crosses the zero-threshold. These frequencies induce a partition of the frequency axis in disjoint sub-bands, where $\boldsymbol{\Psi}(\mathrm{j}\omega)$ is either positive definite or not. This means that, being the Hamiltonian eigenvalues the edges of these sub-bands, the frequency axis is now partitioned in passive and not-passive areas, so that a detailed passivity characterization is available.
In Figure 1.1 we show the described partitioning of the frequency axis in passive and non-passive bands induced by imaginary Hamiltonian eigenvalues. In the left panel we show an eigenvalue of $\mathbf{H}(\mathrm{j}\omega) + \mathbf{H}^{\mathsf{H}}(\mathrm{j}\omega)$ as function of the frequency that, becoming negative, denotes a non-passive frequency band, shown in red. Imaginary Hamiltonian eigenvalues are represented as black dots and bound this violation area. In the right panel we show the Hamiltonian eigen-spectrum, where it is possible to see that the magnitude of purely imaginary eigenvalues coincides with the edges of the violation interval discussed before. The violation bands in the complex plane are represented with red lines.

The main result presented here relies on the strong assumption that $\mathbf{D} + \mathbf{D}^{\mathsf{T}}$ is not singular. However, the same approach can be extended to the case in which $\mathbf{D} + \mathbf{D}^{\mathsf{T}}$ is singular with minor modifications. For details see [21].
In order to relax the non-singularity condition on $\mathbf{D} + \mathbf{D}^{\mathsf{T}}$, it is necessary to slightly modify Theorem 1.3 resulting in an extended eigenvalue problem where, now, no inversions



(a) Passive/non-passive frequency axis partitioning

(b) Violation bands in the Hamiltonian eigenspectrum

Figure 1.1: Frequency axis partitioning induced by Hamiltonian imaginary eigenvalues (immittance systems)

on $\mathbf{D}+\mathbf{D}^{\mathsf{T}}$ are required. The new problem, shown in Equation (1.21) is cast in what is usually called a "generalized eigenvalue problem", where the unknowns are no more the eigenvalues of a matrix, but the ones of a matrix pencil $(\boldsymbol{\mathcal{N}}_0^{ext}, \boldsymbol{\mathcal{K}})$:

$$\boldsymbol{\mathcal{N}}_0^{ext}\mathbf{v} = \mathrm{j}\omega_0\boldsymbol{\mathcal{K}}\mathbf{v} \tag{1.21}$$

where

$$\boldsymbol{\mathcal{N}}_0^{ext} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^{\mathsf{T}} & -\mathbf{C}^{\mathsf{T}} \\ \mathbf{C} & \mathbf{B}^{\mathsf{T}} & \mathbf{D}+\mathbf{D}^{\mathsf{T}} \end{pmatrix}, \qquad \boldsymbol{\mathcal{K}} = \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

This matrix pencil is denoted as "*Skew-Hamiltonian/Hamiltonian*", because $\boldsymbol{\mathcal{N}}_0^{ext}$ has Hamiltonian structure while $\boldsymbol{\mathcal{K}}$ is skew-Hamiltonian.

Up to now, just a state-space realization for $\mathbf{H}(s)$ has been considered. However there are several situations for which a descriptor realization is preferable, e.g., when using MNA (Modified Nodal Analysis) method to automatically solve electrical circuits. For this reason, a generalization of the Hamiltonian approach to descriptor realization is needed. Minor modifications to Theorem 1.3 allow to state that, for immittance systems in descriptor form, the complex frequencies at which passivity violations occur are the purely imaginary generalized eigenvalues of this generalized eigen-problem:

$$\boldsymbol{\mathcal{N}}_0^{ext}\mathbf{v} = \mathrm{j}\omega_0\boldsymbol{\mathcal{K}}\mathbf{v}$$

where

$$\boldsymbol{\mathcal{N}}_0^{ext} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^\mathsf{T} & -\mathbf{C}^\mathsf{T} \\ \mathbf{C} & \mathbf{B}^\mathsf{T} & \mathbf{D}+\mathbf{D}^\mathsf{T} \end{pmatrix}, \qquad \boldsymbol{\mathcal{K}} = \begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^\mathsf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Scattering systems

We now focus on scattering systems.
Recalling Theorem 1.1, a scattering system, to be passive, must have a Bounded Real transfer matrix. Verifying system passivity throughout the complex plane is too expensive in terms of computational effort.
As for the Positive Real Lemma, a formulation of the Bounded Real Lemma exists for rational matrices [2, 21, 44].

Theorem 1.5 *A rational matrix* $\mathbf{H}(s)$ *is Bounded Real if and only if*

1. $\mathbf{H}(s)$ *has no poles in* $\mathbb{C}_+$
2. $\mathbf{H}^*(\mathrm{j}\omega) = \mathbf{H}(-\mathrm{j}\omega)$
3. $\mathbb{I} - \mathbf{H}(\mathrm{j}\omega)^\mathsf{H}\mathbf{H}(\mathrm{j}\omega) \geq 0, \ \forall\omega \in \mathbb{R}$

Conversely from the immittance case, no further conditions are required for purely imaginary poles, since passive scattering systems can not have poles on the imaginary axis. As in Theorem 1.2, the main advantage that the rational nature of the system brings with it, is that Conditions 2 and 3 can be checked just along the imaginary axis. Assuming the system to be asymptotically stable (all the poles of $\mathbf{H}(s)$ has strictly negative real part) and that the state-space realization matrices real, the first two conditions are verified and only the third remains to be checked.
Here, in contrast with the immittance case, a product of transfer matrices appears, so a direct eigenvalues calculation, to guarantee that the smaller one is above the zero threshold, should be avoided. An alternative formulation for Condition 3 is based on the SVD (Singular Values Decomposition) of $\mathbf{H}(\mathrm{j}\omega)$, that reads:

$$\mathbf{H}(\mathrm{j}\omega) = \mathbf{U}(\mathrm{j}\omega)\boldsymbol{\Sigma}(\mathrm{j}\omega)\mathbf{V}(\mathrm{j}\omega)^\mathsf{H}$$

The third condition is then re-formulated in terms of the singular values of $\mathbf{H}(\mathrm{j}\omega)$:

$$\mathbb{I} - \mathbf{H}(\mathrm{j}\omega)^\mathsf{H}\mathbf{H}(\mathrm{j}\omega) \geq 0 \Leftrightarrow \sigma_{max}(\mathbf{H}(\mathrm{j}\omega)) = \|\mathbf{H}(\mathrm{j}\omega)\|_2 \leq 1, \ \forall\omega \in \mathbb{R}$$

Being additionally, by assumption, the transfer matrix $\mathbf{H}(\mathrm{j}\omega)$ regular in an open subset of the complex plane containing the imaginary axis, singular values are continuous functions of $\mathrm{j}\omega$, enabling the use of frequency sampling techniques.

Since any passive system must satisfy the dissipation inequality in (1.15), to derive a precise passivity characterization, it must be particularized for scattering systems.
The supplied power $p(t)$ is:

$$p(t) = \boldsymbol{u}^{\mathsf{T}}\boldsymbol{u} - \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{u}^{\mathsf{T}}\boldsymbol{u} - (\mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u})^{\mathsf{T}}(\mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u})$$

where $\boldsymbol{u}, \boldsymbol{y}$ are, respectively, the input and output signals and the time dependency has been omitted for readability.
The storage function $V(\boldsymbol{x})$, defined as $V(\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}}\mathbf{P}\boldsymbol{x}$, with $\mathbf{P} = \mathbf{P}^{\mathsf{T}} \geq 0$, leads to the following equation:

$$\frac{d}{dt}V(\boldsymbol{x}(t)) = (\mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u})^{\mathsf{T}}\mathbf{P}\boldsymbol{x} + \boldsymbol{x}^{\mathsf{T}}\mathbf{P}(\mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u}) \leq p(t),\ \forall t$$

Combining the previous relation with the dissipation inequality, and splitting the input and state signals, the so-called *Bounded Real Lemma* [2, 34] can be stated.

Lemma 1.2 *A LTI system in scattering form is passive if and only if, for any signal* $\boldsymbol{x}, \boldsymbol{u}$ *satisfying the state equations, it holds that:*

$$\exists \mathbf{P} = \mathbf{P}^{\mathsf{T}} > 0 : \quad \begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{pmatrix}^{\mathsf{T}} \begin{pmatrix}\mathbf{A}^{\mathsf{T}}\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{C}^{\mathsf{T}}\mathbf{C} & \mathbf{P}\mathbf{B} + \mathbf{C}^{\mathsf{T}}\mathbf{D}\\ \mathbf{B}^{\mathsf{T}}\mathbf{P} + \mathbf{D}^{\mathsf{T}}\mathbf{C} & -(\mathbb{I} - \mathbf{D}^{\mathsf{T}}\mathbf{D})\end{pmatrix} \begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{pmatrix} \leq 0$$

In the following we derive, as for immittace representations, a set of results that enables to cast the passivity verification problem in a closed algebraic form. See [21] for details.
Defining $\boldsymbol{\Theta}(s)$ as:

$$\boldsymbol{\Theta}(s) = \mathbb{I} - \mathbf{H}^{\mathsf{H}}(s)\mathbf{H}(s)$$

and denoting the Popov function as:

$$\boldsymbol{\Psi}(s) = \mathbb{I} - \mathbf{H}^{\mathsf{T}}(-s)\mathbf{H}(s)$$

it is easy to see that, when evaluating these functions for $s = \mathrm{j}\omega$, they are equal:

$$\boldsymbol{\Psi}(\mathrm{j}\omega) = \boldsymbol{\Theta}(\mathrm{j}\omega)$$

Passivity condition can be cast in terms of the Popov function as:

$$\boldsymbol{\Psi}(\mathrm{j}\omega) \geq 0,\ \forall \omega \tag{1.22}$$

Equation (1.22) exactly matches the one for immittance representations, where passivity violations are solutions of:

$$\boldsymbol{\Psi}(\mathrm{j}\omega)\boldsymbol{u} = 0 \tag{1.23}$$

for some vector $\boldsymbol{u}$.
To find the zeros of $\boldsymbol{\Psi}(\mathrm{j}\omega)$, a state space realization for $\boldsymbol{\Psi}(s)$ (whose matrices are $\mathbf{A}_\Psi, \mathbf{B}_\Psi, \mathbf{C}_\Psi, \mathbf{D}_\Psi$) is derived, from which it is possible to get a realization for $\boldsymbol{\Psi}^{-1}(s)$, whose purely imaginary poles are the solutions of Equation (1.23). The poles of $\boldsymbol{\Psi}^{-1}(s)$ are the eigenvalues of its state-space dynamic matrix, that reads:

$$\boldsymbol{\mathcal{M}}_1 = \mathbf{A}_\Psi - \mathbf{B}_\Psi \mathbf{D}_\Psi^{-1}\mathbf{C}_\Psi$$

Writing now this matrix in terms the state-space realization matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ of $\mathbf{H}(s)$, we get the following matrix:

$$\boldsymbol{\mathcal{M}}_1 = \begin{pmatrix} \mathbf{A} + \mathbf{B}(\mathbb{I} - \mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{D}^\mathsf{T}\mathbf{C} & \mathbf{B}(\mathbb{I} - \mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{B}^\mathsf{T} \\ -\mathbf{C}^\mathsf{T}(\mathbb{I} - \mathbf{D}\mathbf{D}^\mathsf{T})^{-1}\mathbf{C} & -\mathbf{A}^\mathsf{T} - \mathbf{C}^\mathsf{T}\mathbf{D}(\mathbb{I} - \mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{B}^\mathsf{T} \end{pmatrix}$$

Matrix $\boldsymbol{\mathcal{M}}_1$ has Hamiltonian structure, since it satisfies the condition in (1.20).

What relates matrix $\boldsymbol{\mathcal{M}}_1$ with system passivity is given by the following theorem: [4,17,21]

Theorem 1.6 *Let $\mathbf{H}(s)$ be the transfer matrix of a scattering system, whose state space matrices are $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, where $\mathbf{A}$ has no purely imaginary eigenvalues and $\mathbb{I} - \mathbf{D}^\mathsf{T}\mathbf{D}$ is non-singular. Then, $\mathrm{j}\omega_0$ is an eigenvalue of $\boldsymbol{\mathcal{M}}_1$ if and only if 0 is an eigenvalue of $\boldsymbol{\Psi}(\mathrm{j}\omega_0)$ and 1 a singular value of $\mathbf{H}(\mathrm{j}\omega_0)$.*

This result allows us to derive the following theorem, that provides a sufficient passivity condition for scattering systems:

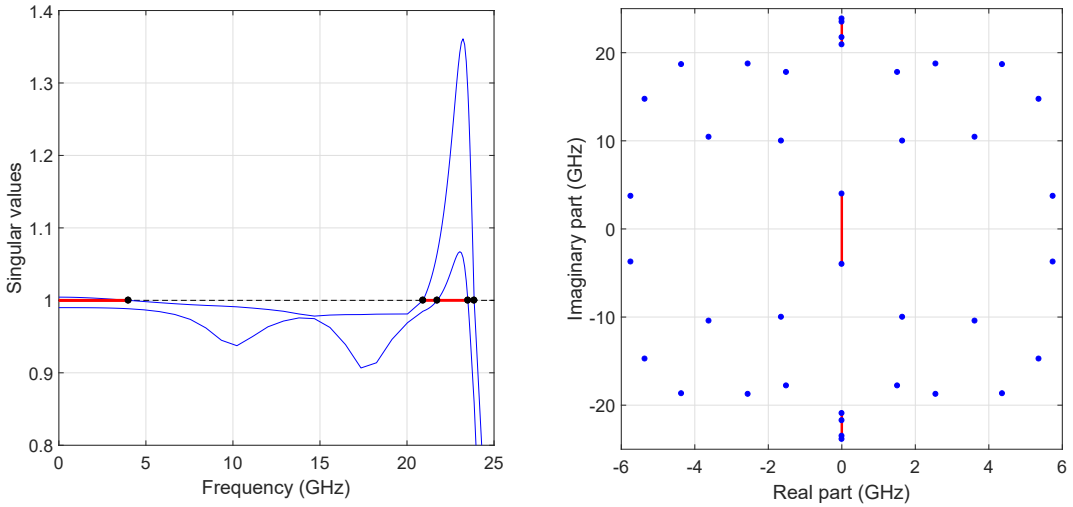
Theorem 1.7 *Let $\mathbf{H}(s)$ be the transfer function of an asymptotically passive ($\|\mathbf{D}\|_2 < 1$) and stable scattering system, whose state-space matrices are ($\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$). The system is uniformly passive if $\boldsymbol{\mathcal{M}}_1$ has no purely imaginary eigenvalues*

Furthermore, the frequencies ω_i solving $\boldsymbol{\Psi}(\mathrm{j}\omega_i)\boldsymbol{u} = 0$, i.e., the Hamiltonian imaginary eigenvalues, induce a partition of the frequency axis in passive and not-passive sub-bands. These considerations allow to characterize in details the passivity of a system for any frequency value.
Figure 1.1 shows the partitioning of the frequency axis in passive and non-passive bands induced by imaginary Hamiltonian eigenvalues. In the left panel we show singular values of $\mathbf{H}(\mathrm{j}\omega)$ that, when their value exceeds one, allow to identify non-passive regions, represented in red. Imaginary Hamiltonian eigenvalues are represented as black dots and bound these violation areas. In the right panel we show the Hamiltonian eigenspectrum, where we see that the magnitude of purely imaginary eigenvalues coincide with the edges of the violations interval discussed before. The violations bands in the complex plane are represented with red lines.
As we did for immittance systems, it is possible to relax the non-singularity condition on $\mathbb{I} - \mathbf{D}^\mathsf{T}\mathbf{D}$.
As proposed in [21], Theorem 1.6 can be generalized to the case in which $\mathbf{D}$ is arbitrary. Slightly modifying its proof, it is possible to define an extended eigen-problem shown in (1.24), that does not require any inversion of $\mathbb{I} - \mathbf{D}\mathbf{D}^\mathsf{T}$ and $\mathbb{I} - \mathbf{D}^\mathsf{T}\mathbf{D}$, as:

$$\boldsymbol{\mathcal{M}}_1^{ext}\mathbf{v} = \mathrm{j}\omega_0\boldsymbol{\mathcal{K}}\mathbf{v} \tag{1.24}$$

where

$$\boldsymbol{\mathcal{N}}_0^{ext} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}^\mathsf{T} & \mathbf{0} & -\mathbf{C}^\mathsf{T} \\ \mathbf{0} & \mathbf{B}^\mathsf{T} & -\mathbb{I} & \mathbf{D}^\mathsf{T} \\ \mathbf{C} & \mathbf{0} & \mathbf{D} & \mathbb{I} \end{pmatrix}, \qquad \boldsymbol{\mathcal{K}} = \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$



(a) Passive/non-passive frequency axis partitioning

(b) Violation bands in the Hamiltonian eigenspectrum

Figure 1.2: Frequency axis partitioning induced by Hamiltonian imaginary eigenvalues (scattering systems)

It can be proven that purely imaginary eigenvalues of (1.24) correspond exactly to the location on the frequency axis of passivity violations.

Previous results are based on the assumption that a state-space realization for $\mathbf{H}(s)$ is used. Here, we provide a generalization of the Hamiltonian-driven passivity characterization to descriptor realizations, that will be used extensively later on in this work, and are of paramount importance in many other applications.
Passivity violations are again defined by complex frequencies $\mathrm{j}\omega_i$ for which $\mathbf{\Psi}(\mathrm{j}\omega_i)\mathbf{v} = 0$. Suitably modifying Theorem 1.6, we find that this condition is reached if and only if $\mathrm{j}\omega_i$ is an eigenvalue of the generalized eigenproblem in (1.25).

$$\boldsymbol{\mathcal{M}}_1^{ext}\mathbf{v} = \mathrm{j}\omega_0\boldsymbol{\mathcal{K}}\mathbf{v} \tag{1.25}$$

where

$$\boldsymbol{\mathcal{N}}_0^{ext} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}^\mathsf{T} & \mathbf{0} & -\mathbf{C}^\mathsf{T} \\ \mathbf{0} & \mathbf{B}^\mathsf{T} & -\mathbb{I} & \mathbf{D}^\mathsf{T} \\ \mathbf{C} & \mathbf{0} & \mathbf{D} & \mathbb{I} \end{pmatrix}, \qquad \boldsymbol{\mathcal{K}} = \begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^\mathsf{T} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Chapter 2

Multivariate Macromodels

This chapter is co-authored by T. Bradde, M. De Stefano and A. Zanco.

In the previous chapter, we assumed that the system under modeling is characterized by a fixed (yet unknown) physical structure. In many situations, however, this hypothesis is not the most suitable: some physical parameters of the system could be design objectives or could be intrinsically uncertain due to production process tolerances.
A parametric macromodel is able to reproduce the system behaviour for all the possible values that the varying parameters assume within a prescribed range. This possibility proves to be extremely useful in many fields of the design process, from the optimization of the design variables, to the simulation of worst-case scenarios induced by the physical realization of the structure. Typical examples regard the role of temperature in electronic devices, the geometrical parameters of an interconnect, the linearization point of a nonlinear device, and many more.
The construction flow of a parametric macromodel requires the knowledge of the input-output behavior for a discrete number of values within the range that the parameters can span; once those data are collected and processed, the interpolation algorithm returns a closed form description of the system within the entire range of variation.
In this case, the input-output data must be representative of the model behavior within all the range of values assumed by each parameter; in particular, consider the case in which the model is required to depend on a number ρ of design parameters. Then, for the i-th parameter we can denote its variation range as:

$$\Theta_i = \left[\vartheta^i_{min},\ \vartheta^i_{max}\right] \quad for \quad i = 1,2,\ldots,\rho$$

Thus, the global parameter domain can be defined as:

$$\Theta = \Theta_1 \times \Theta_2 \times \cdots \times \Theta_\rho \quad \subseteq \mathbb{R}^\rho \tag{2.1}$$

In a discretized parameters space, a point in Θ is uniquely identified by its projections along the parameters axes. To keep the notation compact, this point is denoted as

$$\vartheta_{\mathbf{m}} = [\vartheta_{m_1}, \ldots, \vartheta_{m_\rho}]^\mathsf{T}$$

where $\mathbf{m}$ is a multi-index $\mathbf{m} = [m_1, \ldots, m_\rho]$.

To synthetize a parametric macromodel, a set of M points in the parameter domain Θ is defined to be representative of the parametric system response; for each of these points, we collect K frequency samples of the transfer functions associated with the underlying system. The resulting dataset reads:

$$\breve{\mathbf{H}}_{k,m} = \breve{\mathbf{H}}(s_k; \vartheta_m) \quad for \quad k = 1,2,\ldots,K \quad m = 1,2,\ldots,M$$

If, as it is common, we collect data at real frequencies ω_k, our goal is to obtain a model:

$$\mathbf{H}(\mathrm{j}\omega; \vartheta) \approx \breve{\mathbf{H}}(\mathrm{j}\omega; \vartheta) \quad for \quad \vartheta \in \Theta, \quad \omega \in [\omega_{min}, \omega_{max}]$$

While the structure of an univariate model is supposed to be a rational function of the Laplace variable, we are free to cast the dependence of the model on the parameters in a larger set of possible structures: a variety of basis functions can be used to fit the data. The thesis project is particularly focused on the investigation of issues related to the construction of precise and reliable parametric macromodels, for which many open problems still exist.

2.1 Parametric Model Formulation

Approximating the true system response $\breve{\mathbf{H}}(s_k; \vartheta_k)$ in a suitable macromodel form is fundamental to include the curve fitting result in system-level simulations using standard circuit solvers such as SPICE. Several mathematical structures are available: the identification algorithm efficiency, in frequency and time domain, is affected by this choice. Moreover, all the formulations may suffer from ill-conditioning depending on the parameter-dependent basis choice.

Therefore, considering a P-ports multivariate macromodel of a generic LTI system, we adopt the so-called *Parameterized Sanathanan-Koerner (PSK)* form [37], [36], [13], [12], [22]

$$\mathbf{H}(s; \vartheta) = \frac{\mathbf{N}(s, \vartheta)}{\mathsf{D}(s, \vartheta)} = \frac{\sum_{n=0}^{\bar{n}} \sum_{\ell=1}^{\bar{\ell}} \mathbf{R}_{n,\ell}\, \xi_\ell(\vartheta)\, \varphi_n(s)}{\sum_{n=0}^{\bar{n}} \sum_{\ell=1}^{\bar{\ell}} r_{n,\ell}\, \xi_\ell(\vartheta)\, \varphi_n(s)}. \tag{2.2}$$

We remark that the model numerator and denominator are constructed by linear combination of suitable basis functions: it is straightforward to prove that if the basis functions $\varphi_n(s)$ are rational, the model indicated in (2.2) is a rational function $\forall \vartheta$.
In particular, we denoted with $\bar{n}$ the frequency basis order and with $\bar{\ell}$ the cardinality of the parameter-dependent basis function. To maintain the notation compact, we define a multi-index $\ell = (\ell_1, \ldots, \ell_\rho)$, if $\rho > 1$.
Both the numerator and denominator coefficients are guaranteed real-valued: they are indicated with $\mathbf{R}_{n,\ell} \in \mathbb{R}^{P \times P}$ and $r_{n,\ell} \in \mathbb{R}$, respectively, in (2.2).

We can simplify the model expression presented in (2.2), gathering the parameter information

$$\mathbf{H}(s; \vartheta) = \frac{\mathbf{N}(s, \vartheta)}{\mathsf{D}(s, \vartheta)} = \frac{\sum_{n=0}^{\bar{n}} \mathbf{R}_n(\vartheta)\, \varphi_n(s)}{\sum_{n=0}^{\bar{n}} r_n(\vartheta)\, \varphi_n(s)}, \tag{2.3}$$

where

$$\mathbf{R}_n(\vartheta) = \sum_{\ell_N=1}^{\bar{\ell}_N} \mathbf{R}_{n,\ell_N}\, \xi_{\ell_N}(\vartheta) \quad r_n(\vartheta) = \sum_{\ell_D=1}^{\bar{\ell}_D} r_{n,\ell_D}\, \xi_{\ell_D}(\vartheta) \tag{2.4}$$

are the numerator and denominator model coefficients, respectively.
Note that a different parameter-dependent basis order for numerator($\bar{\ell}_N$) and denominator($\bar{\ell}_D$) polynomials is possible, as specified in (2.4). Without loss of generality, in the following we will set $\bar{\ell}_N = \bar{\ell}_D = \bar{\ell}$.
The model structure presented before is completely general with respect to the input data set $\breve{\mathbf{H}}(s_k; \vartheta_k)$ representation (scattering, admittance or impedance).

2.1.1 Parameter-dependent basis functions

The variations induced by the external parameters $\vartheta \in \Theta$ are embedded in the model structure (2.2) through the parameter-dependent basis functions $\xi_\ell(\vartheta)$. These basis functions must be selected carefully because upon this choice depends the fitting accuracy.
The literature offers several sets of functions, which are characterized by their own numerical properties.
In the following we will consider only one external parameter ($\rho = 1$).
One important point for our further observations is the (commonly used) procedure of improving the numerical conditioning of fitting algorithms by the normalization of the polynomials argument within $[-1,\ 1]$. In particular, we compute the normalized parameter value $\tilde{\vartheta}$ as:

$$\tilde{\vartheta} = -1 + 2 \cdot \frac{\vartheta - \vartheta_{min}}{\vartheta_{max} - \vartheta_{min}}$$

The problem conditioning will direct affect the parameter-dependent basis choice.
We now provide several examples of the available choices for the parameter-dependent basis.

Monomials

The simplest function that could be used to capture the parameter evolution is defined as the standard monomials basis functions [41]:

$$\xi_\ell(\vartheta) = \vartheta^\ell$$

where $\ell = 0, ..., \bar{\ell}$ (as defined in (2.2)).
We provide a numerical example, realizing a third-order basis as showed in Fig.2.1.
This represents the most intuitive case for parameter-dependent basis definition, but this sort of basis function usually leads to the construction of an *ill-conditioned* fitting problem.

Chebychev Polynomials

Other functions that are widely used as parameter bases are the orthogonal polynomials [1], [28]. From [7], we know that any orthogonal polynomial can be expressed with this

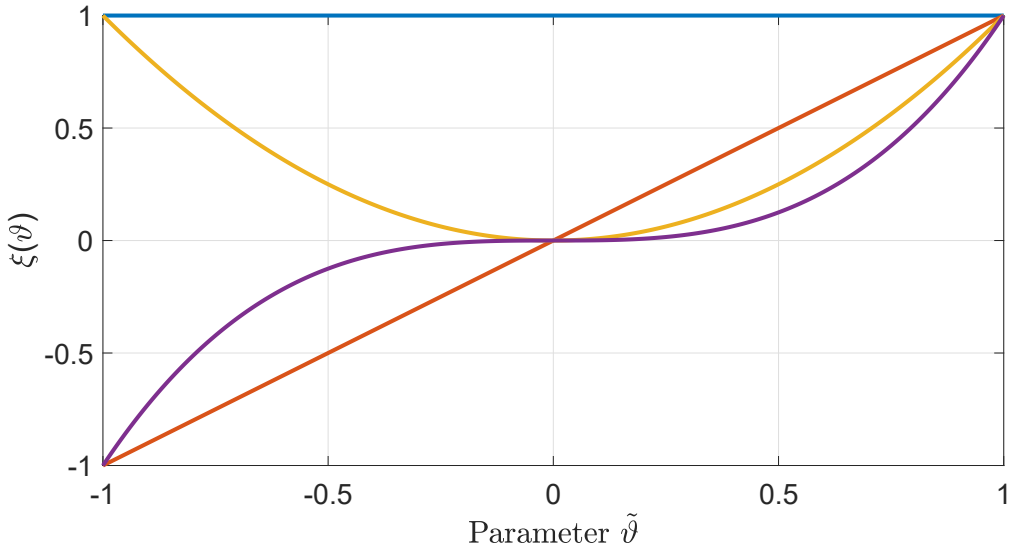


Figure 2.1: Monomials parameter-dependent basis evolution for $\ell = 0,1,2,3$.

recurrence relation:

$$\xi_{\ell+1}(\vartheta) = (\alpha_\ell \vartheta + \beta_\ell)\xi_\ell(\vartheta) + \delta_{\ell-1}\xi_{\ell-1}(\vartheta)$$

In the following we will extensively use Chebychev polynomials, a special class of orthogonal polynomials, for which the expansion coefficients α, β, δ are equal to:

$$\begin{aligned} &\alpha_0 = 1\,, && \beta_0 = 0\,, && \delta_0 = 0, && \\ &\alpha_\ell = 2\,, && \beta_\ell = 0\,, && \delta_\ell = -1 && \forall \ell \geq 1 \end{aligned}$$

It is well known that the basis functions defined as before present very favourable numerical properties, which lead to a well-conditioned (and easy manageable) fitting problem with a reasonable condition number. We denote the Chebychev polynomials of the first kind basis functions $\xi_\ell(\vartheta) = T_\ell(\vartheta)$ (see [3] and [15]) as:

$$T_\ell(\vartheta) = \cos[\ell \cos^{-1}(\vartheta)]\,, \quad \vartheta \in [-1,\ 1]\,, \quad \ell = 0, ..., \bar{\ell}$$

which is equivalent to the standard expression:

$$T_\ell(\cos t) = \cos(\ell t)\,, \quad t \in [0,\ 2\pi]\,, \quad \ell = 0, ..., \bar{\ell}$$

An example of the fourth order Chebychev polynomials (first kind) is reported in Fig. 2.2.

Fourier Series

In order to guarantee a parameterization from a smooth function, when ϑ implies periodic variations, with $\vartheta \in [0,\ 2\pi]$ (e.g. the external parameter is an angle), as discussed in [20], we can define a parameter-dependent basis function as the standard Fourier basis in the

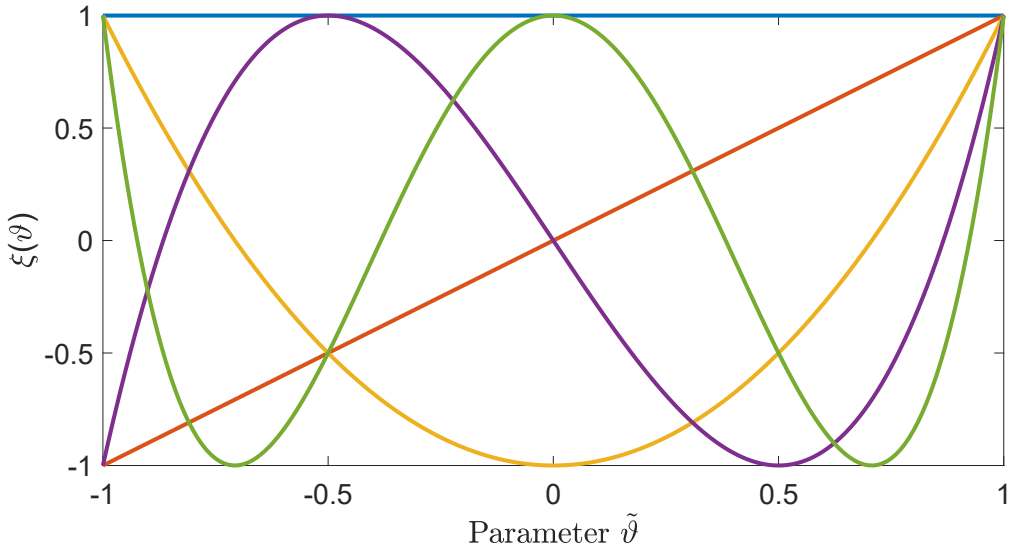


Figure 2.2: Chebychev parameter-dependent basis evolution for $\ell = 0,1,2,3$.

trigonometric form:

$$\xi_\ell(\vartheta) = \begin{cases} 1, & \ell = 0 \\ \cos(\lceil \ell/2 \rceil \vartheta), & \ell = 1,3,5,\ldots \\ \sin(\lceil \ell/2 \rceil \vartheta), & \ell = 2,4,6,\ldots \end{cases}$$

where the argument of $\lceil \cdot \rceil$ is rounded to the nearest larger integer. Figure 2.3 provides a numerical example for the first five terms of the Fourier basis ($\ell = 0,...,4$), through the parameter range $\vartheta \in [0°,\ 360°]$.

2.1.2 State Space and Descriptor Forms

We now present the state-space and descriptor realizations of a parameter-dependent LTI system, starting from the pole-residue form of the model $\mathbf{H}(s;\vartheta)$. As in the univariate case, also for a multivariate model this representation is appropriate to describe the properties of the model in algebraic form.

State Space Realizations

Following the procedure reported in Section 1.4.2, we can realize a parameter-dependent macromodel equivalent state-space description. Recalling the pole-residue model form of (1.5) and embedding the parameter dependency, the extension to the parametric case is straightforward. In fact, considering now the model in (2.3), with $\varphi_n(s)$ defined as the

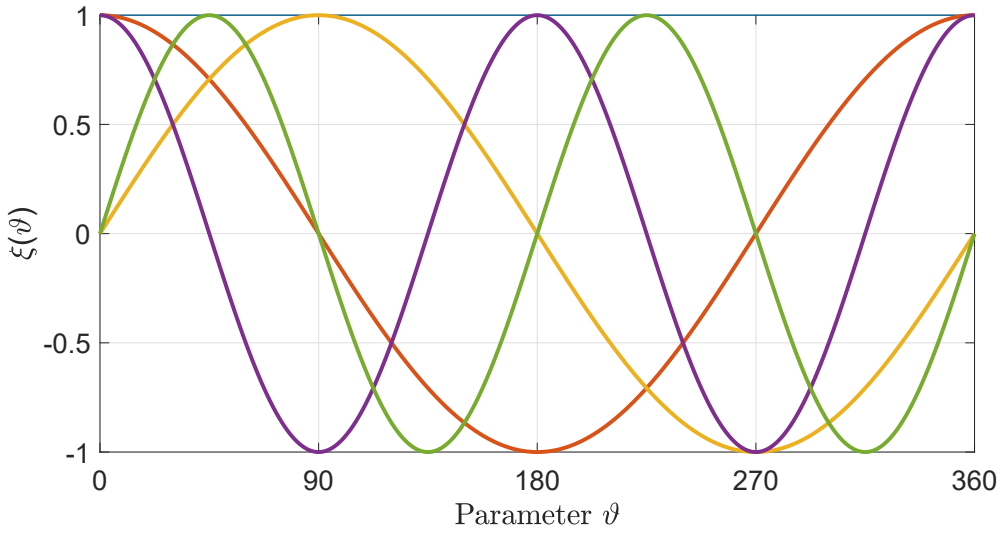


Figure 2.3: First four terms ($\ell = 0,1,2,3$) of the Fourier parameter-dependent basis, through the parameter range $\vartheta \in [0°,360°]$. The polynomials arguments is normalized within $[-1,1]$ using the variable range.

partial-fraction basis with a prescribed set of poles q_n, (see Section 1.2.1), we can write:

$$\mathbf{N}(s,\vartheta) = \mathbf{R}_0(\vartheta) + \sum_{n=1}^{\bar{n}} \frac{\mathbf{R}_n(\vartheta)}{s - q_n}$$
$$\mathsf{D}(s,\vartheta) = r_0(\vartheta) + \sum_{n=1}^{\bar{n}} \frac{r_n(\vartheta)}{s - q_n}$$

which allows us to construct the two separate state-space realizations for $\mathbf{N}(s,\vartheta)$ and $\mathsf{D}(s,\vartheta)$ as:

$$\mathbf{N}(s,\vartheta) \leftrightarrow \left(\begin{array}{c|c} \mathbf{A}_0 & \mathbf{B}_0 \\ \hline \mathbf{C}_1(\vartheta) & \mathbf{D}_1(\vartheta) \end{array}\right)$$
$$\mathsf{D}(s,\vartheta)\mathbb{I}_P \leftrightarrow \left(\begin{array}{c|c} \mathbf{A}_0 & \mathbf{B}_0 \\ \hline \mathbf{C}_2(\vartheta) & \mathbf{D}_2(\vartheta) \end{array}\right) \tag{2.5}$$

where

$$\begin{aligned}
\mathbf{A}_0 &= \text{blkdiag}\{\mathbf{A}_{0r}, \mathbf{A}_{0c}\} \\
\mathbf{B}_0^{\mathsf{T}} &= [\mathbf{B}_{0r}^{\mathsf{T}}, \mathbf{B}_{0c}^{\mathsf{T}}] \\
\mathbf{C}_1(\vartheta) &= \begin{bmatrix} \mathbf{R}_1(\vartheta) & \cdots & \mathbf{R}_{\bar{n}}(\vartheta) \end{bmatrix} \\
\mathbf{C}_2(\vartheta) &= \begin{bmatrix} \mathbb{I}_P\, r_1(\vartheta) & \cdots & \mathbb{I}_P\, r_{\bar{n}}(\vartheta) \end{bmatrix} \\
\mathbf{D}_1(\vartheta) &= \mathbf{R}_0(\vartheta) \\
\mathbf{D}_2(\vartheta) &= \mathbb{I}_P\, r_0(\vartheta).
\end{aligned}$$

with

$$\begin{aligned}\mathbf{A}_{0r} =&\text{blkdiag}\{q_n\mathbb{I}_P\}_{n=1}^{\bar{n}_r}\\ \mathbf{A}_{0c} =&\text{blkdiag}\left\{\begin{bmatrix} q_n'\mathbb{I}_P & q_n''\mathbb{I}_P\\ -q_n''\mathbb{I}_P & q_n'\mathbb{I}_P\end{bmatrix}\right\}_{n=1}^{\bar{n}_c}\\ \mathbf{B}_{0r} =&\left[1,\ldots,1\right]^{\mathsf{T}}\otimes\mathbb{I}_P\\ \mathbf{B}_{0c} =&\left[2,0,\ldots,2,0\right]^{\mathsf{T}}\otimes\mathbb{I}_P\end{aligned}$$

Following the steps described in [36], we finally obtain a compact model state-space realization cascading the expressions in (2.5) as:

$$\mathbf{H}(s;\vartheta)=\mathbf{N}(s,\vartheta)(\mathsf{D}(s,\vartheta)^{-1}\mathbb{I}_P)\leftrightarrow\left(\begin{array}{c|c}\mathbf{A}_0-\mathbf{B}_0\mathbf{D}_2^{-1}(\vartheta)\mathbf{C}_2(\vartheta) & \mathbf{B}_0\mathbf{D}_2^{-1}(\vartheta)\\ \hline \mathbf{C}_1(\vartheta)-\mathbf{D}_1(\vartheta)\mathbf{D}_2^{-1}(\vartheta)\mathbf{C}_2(\vartheta) & \mathbf{D}_1(\vartheta)\mathbf{D}_2^{-1}(\vartheta)\end{array}\right)$$

We recall [36] for more details.

Descriptor Forms

Recalling to the descriptor representation (1.8) in Section 1.4.2, we now define its parameter-dependent form [36] as:

$$\begin{aligned}\mathbf{E}=\begin{pmatrix}\mathbb{I}_N & \mathbf{0}_{N,P}\\ \mathbf{0}_{P,N} & \mathbf{0}_{P,P}\end{pmatrix}\quad \mathbf{A}(\vartheta)=\begin{pmatrix}\mathbf{A}_0 & \mathbf{B}_0\\ \mathbf{C}_2(\vartheta) & \mathbf{D}_2(\vartheta)\end{pmatrix}\\ \mathbf{C}(\vartheta)=\begin{pmatrix}\mathbf{C}_1(\vartheta) & \mathbf{D}_1(\vartheta)\end{pmatrix}\quad \mathbf{B}=\begin{pmatrix}\mathbf{0}_{N,P}\\ -\mathbb{I}_P\end{pmatrix}\end{aligned}$$

with $\mathbf{0}_{J,K}$ the null matrix of size $J\times K$.
The model expression of (2.2) is equivalent to:

$$\mathbf{H}(s;\vartheta)=\mathbf{C}(\vartheta)(s\mathbf{E}-\mathbf{A}(\vartheta))^{-1}\mathbf{B}$$

as detailed in [36] and [18].

2.2 Parametric Model Extraction

Using the model shown in Equation (2.2), to find the residues $\mathbf{R}_{n,\ell}$, $r_{n,\ell}$ we solve the following problem:

$$\left\|\mathbf{H}(\mathrm{j}\omega_k;\vartheta)-\breve{\mathbf{H}}_{k,m}\right\|^2\approx 0 \tag{2.6}$$

By construction, the problem in (2.6) is non-linear, thus we proceed with a linear relaxation that brings to an iterative formulation, known as *Parameterized Sanathanan-Koerner* iteration (PSK), a parametric extension of the GSK iteration shown in Section

1.3.1. For details see [10, 36, 40].
We set-up the PSK scheme as:

$$\min \left\| \frac{\mathbf{N}^{\mu}(\mathrm{j}2\pi f_k, \vartheta_m) - \mathsf{D}^{\mu}(\mathrm{j}2\pi f_k, \vartheta_m)\, \breve{\mathbf{H}}_{k,m}}{\mathsf{D}^{\mu-1}(\mathrm{j}2\pi f_k, \vartheta_m)} \right\|_F^2$$
$$\mu = 1, 2, \ldots$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The iterative process is initialized with $\mathsf{D}^0(\mathrm{j}\omega_k, \vartheta_m) = 1$ and stops when the estimates of model coefficients $\mathbf{R}_{n,\ell}$, $r_{n,\ell}$ stabilize.
With respect to the non-parametric case (GSK), the PSK least-square problem is a computationally more expensive task, due to the higher dimension of the regressor matrix. To keep estimation times reasonable, there are advanced PSK formulation, known as *Fast Parameterized Sanathanan-Koerner, (FPSK)*, that alleviate this problem [5, 19].

Chapter 3

Non-Parameterized Passivity Enforcement

3.1 Model Perturbation to Enforce Passivity

In this chapter the problem of enforcing the passivity of a lumped LTI system will be addressed. With passivity enforcement we refer to the procedure through which a non-passive model is forced to be passive by slightly perturbing its coefficients. This can be achieved thanks to passivity conditions derived in Section 1.6 that, in addition to characterizing the passivity of immittance and scattering models, are the starting point to formulate proper constraints for passivity enforcement algorithms. Here we will briefly go through some methods that allow, starting from passivity violations locations, to enforce the model to be passive, still keeping its responses accurate as much as possible.
Indicating the non-passive model transfer matrix as $\mathbf{H}(s)$ and the passive one as $\widehat{\mathbf{H}}(s)$, we define the difference $\Delta\mathbf{H}(s) = \widehat{\mathbf{H}}(s) - \mathbf{H}(s)$. Our aim is to perturb the non-passive model transfer matrix $\mathbf{H}(s)$ toward $\widehat{\mathbf{H}}(s)$. To get the proper coefficients variation, several different frameworks has been developed. In fact, it is possible to directly perturb, under suitable constraints, the model residues while minimizing the error with respect to data but, studying the problem from other standpoints, the realization matrices or even the Hamiltonian matrix associated with the system may be used to find proper model perturbations. In this work we focus just on the model residues-perturbation method for local passivity enforcement. For more details about this and the other schemes used in enforcing macro-model passivity, see [21].

We recall here the non-parameterized rational model structure:

$$\mathbf{H}(s) = \frac{\mathbf{N}(s)}{\mathrm{D}(s)} = \frac{\sum_{n=0}^{\bar{n}} \mathbf{R}_n \ \varphi_n(s)}{\sum_{n=0}^{\bar{n}} r_n \ \varphi_n(s)} \tag{3.1}$$

Potentially both numerator and denominator residues can be perturbed. However, a perturbation of denominator coefficients r_n, may have detrimental effects on the model in terms of stability and frequency response. For this reason, the only degrees of freedom we will exploit are the numerator residues $\mathbf{R}_n$. This consideration leads to define explicitly

the difference transfer matrix $\Delta\mathbf{H}(s)$ as:

$$\Delta\mathbf{H}(s) = \frac{\sum_{n=0}^{\bar{n}} \Delta\mathbf{R}_n \ \varphi_n(s)}{\sum_{n=0}^{\bar{n}} r_n \ \varphi_n(s)}$$

where $\Delta\mathbf{R}_n$ corresponds to the perturbation on the n-th numerator residue. Once the perturbations $\Delta\mathbf{R}_n$ for $n = 0, \ldots, \bar{n}$ are known, the perturbed model $\widehat{\mathbf{H}}(s)$ can be constructed as:

$$\widehat{\mathbf{H}}(s) = \frac{\widehat{\mathbf{N}}(s)}{\mathsf{D}(s)} = \frac{\sum_{n=0}^{\bar{n}} (\mathbf{R}_n + \Delta\mathbf{R}_\mathbf{n}) \ \varphi_n(s)}{\sum_{n=0}^{\bar{n}} r_n \ \varphi_n(s)}$$

In the next sections we address the problem of determining these perturbations that must ensure the final model $\widehat{\mathbf{H}}(s)$ to be passive and accurate.

3.2 A Local Enforcement Approach

3.2.1 Asymptotic passivity enforcement

Before discussing in details how to find residues perturbations that guarantee passivity and accuracy for finite frequencies, some remarks are in order about the asymptotic passivity enforcement.
The model we must consider in this case is simpler with respect to the more general one in Equation (3.1) because all the summation terms vanish when $\omega \to \infty$, except for $\mathbf{R}_0$ and r_0 since related to the unitary partial fraction basis term $\varphi_0(s) = 1$.
We define $\mathbf{H}(\infty)$ as:

$$\mathbf{H}(\infty) = \frac{\mathbf{R}_0}{r_0}, \ \mathbf{H}(\infty) \in \mathbb{R}^{P,P}$$

To ensure asymptotic passivity, recalling Theorems 1.2 and 1.5, it must hold that:

$$\mathbf{H}(\infty) + \mathbf{H}^{\mathsf{H}}(\infty) = \mathbf{H}(\infty) + \mathbf{H}^{\mathsf{T}}(\infty) \geq 0 \tag{3.2}$$

for immittance systems, and:

$$\mathbb{I} - \mathbf{H}^{\mathsf{H}}(\infty)\mathbf{H}(\infty) = \mathbb{I} - \mathbf{H}^{\mathsf{T}}(\infty)\mathbf{H}(\infty) \geq 0 \tag{3.3}$$

for scattering systems.

In the following, a simple yet effective asymptotic enforcement technique is detailed for immittance and scattering systems.

Immittance systems asymptotic perturbation

Condition in (3.2) can be re-written as:

$$\frac{\mathbf{H}(\infty) + \mathbf{H}^{\mathsf{T}}(\infty)}{2} = \mathbf{H}^{'}(\infty) \geq 0$$

where $\mathbf{H}'(\infty)$ is the symmetric part of $\mathbf{H}(\infty)$. This result states that any perturbation on the skew-symmetric part of $\mathbf{H}(\infty)$ is ineffective in terms of asymptotic passivity. For this reason, from now on, only the symmetric part will be considered.
Any real symmetric matrix admits a spectral decomposition, thus $\mathbf{H}'(\infty)$ can be written as:

$$\mathbf{H}'(\infty) = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{T}}$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_P)$ contains in its main diagonal the eigenvalues of $\mathbf{H}'(\infty)$ and $\mathbf{V}$ contains the eigenvectors of $\mathbf{H}'(\infty)$ in its columns.

The model results to be non-asymptotically passive if there is at least one eigenvalue $\lambda_i < 0$. Assuming that the eigenvalues in $\mathbf{\Lambda}$ are sorted in descending order ($\lambda_1 \geq \lambda_2, \ldots, \geq \lambda_P$), asymptotic-passivity can be enforced simply by replacing negative eigenvalues with a positive real constant ϵ while keeping fixed the other ones.
If i is the index for the last non-negative eigenvalue, we can formulate this passivity enforcement method as:

$$\widehat{\lambda}_j = \begin{cases} \lambda_j, & j = 1, \ldots, i \\ \epsilon, & j = i+1, \ldots, P. \end{cases}$$

Once the eigenvalues $\widehat{\lambda}_i$ are known, the perturbed matrix $\widehat{\mathbf{H}}'(\infty)$ is reconstructed as:

$$\widehat{\mathbf{H}}'(\infty) = \mathbf{V}\widehat{\mathbf{\Lambda}}\mathbf{V}^{\mathsf{T}}$$

where $\widehat{\mathbf{\Lambda}} = \text{diag}(\widehat{\lambda}_1, \ldots, \widehat{\lambda}_P)$.

Once the perturbed residue $\widehat{\mathbf{R}}_0$ is known, an asymptotically passive model can be derived just by replacing $\mathbf{R}_0$ with $\widehat{\mathbf{R}}_0$.
This perturbation scheme is optimal, since its energy $\|\Delta\mathbf{H}(\infty)\|_2^2$ coincides with the minimum amount necessary to "lift" the negative eigenvalues up to the prescribed value.

Scattering systems asymptotic perturbation

Condition in (3.3) can be re-stated in terms of singular values of $\mathbf{H}(\infty)$ as:

$$\mathbb{I} - \mathbf{H}^{\mathsf{T}}(\infty)\mathbf{H}(\infty) \geq 0 \Leftrightarrow \max_{i=1,\ldots,P} \sigma_i(\mathbf{H}(\infty)) \leq 1$$

Similarly to what has been done for immittance systems, asymptotic passivity enforcement is fulfilled by singular values truncation, where non-passive singular values ($\sigma_i > 1$) are replaced with passive ones ($\sigma_i \leq 1$).
Since $\mathbf{R}(\infty)$ is a real valued matrix, its singular value decomposition is:

$$\mathbf{H}(\infty) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}}$$

where $\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_P)$ and $\mathbf{U}, \mathbf{V}$ collect, respectively, left and right singular vectors, it is easy to understand that it is sufficient to modify the diagonal elements on $\mathbf{\Sigma}$.

By construction, the singular values in $\mathbf{\Sigma}$ are sorted in descending order, thus the model is non-asymptotically passive if there exist $i \in \{1, \dots, P\}$ such that $\sigma_j > 1$ for $j \in \{1, \dots, i\}$.

To enforce passivity is then sufficient to define a new set of truncated singular values as:

$$\widehat{\sigma}_j = \begin{cases} 1 - \epsilon, & j = 1, \dots, i \\ \sigma_j, & j = i+1, \dots, P. \end{cases}$$

with $0 < \epsilon < 1$, such that $\widehat{\sigma}_1 \leq 1$.

The perturbed matrix $\widehat{\mathbf{H}}(\infty)$ is then re-constructed starting from the new passive singular values as

$$\widehat{\mathbf{H}}(\infty) = \mathbf{U}\widehat{\mathbf{\Sigma}}\mathbf{V}^{\mathsf{T}}$$

where $\widehat{\mathbf{\Sigma}} = \mathrm{diag}(\widehat{\sigma}_1, \dots, \widehat{\sigma}_P)$.
The asymptotically passive model is now obtained by replacing the numerator residue $\mathbf{R}_0$ with the perturbed one $\widehat{\mathbf{R}}_0$.
Also this perturbation is optimal, since it is the smallest one allowing to bring the largest singular value σ_1 below the prescribed threshold $1 - \epsilon$.

This is not the unique method that allows to enforce the asymptotic model passivity. In fact, through a local approach, it is possible to formulate constraints at infinite frequency to be used inside an optimization problem. This last method is the one we use in the following.
In the next section we are going to detail how to derive local passivity constraints and how to use them during an optimization problem to enforce the model passivity.

3.2.2 Local passivity constraints

In Section 3.1, the problem of enforcing the passivity of lumped LTI models has been introduced. We stated that our aim is to enforce the passivity while minimizing the deviation of the model response from data. This is equivalent to minimizing the perturbation energy induced by $\Delta\mathbf{H}(s)$ while imposing passivity with suitable constraints. The passivity enforcement procedure can be thus stated in terms of an optimization problem [21]:

$$\begin{aligned} &\min \|\Delta\mathbf{H}(s)\|^2 \\ &\text{s.t. } \mathbf{H}(s) + \Delta\mathbf{H}(s) \text{ is passive.} \end{aligned} \tag{3.4}$$

where the cost-function is the perturbation energy, considered as some norm of the transfer matrix perturbation.
The cost function $\|\Delta\mathbf{H}(s)\|^2$ can be cast in a form suitable for a quadratic convex optimization problem, where the decision variables are explicit. To this end, considering $s = \mathrm{j}\omega_k$, it is possible to rewrite the cost-function $\|\Delta\mathbf{H}\|^2$ as $\mathcal{E}^2$, where:

$$\mathcal{E}^2 = \sum_{i,j}^{P} \mathcal{E}_{i,j}^2, \text{ with } \mathcal{E}_{i,j}^2 = \sum_{k=1}^{\bar{k}} |\Delta H_{i,j}(\mathrm{j}\omega_k)|^2$$

where k is a frequency sample index and $(i,\ j)$ indicates the i-th, j-th entry of the matrix $\Delta\mathbf{H}(\mathrm{j}\omega_k)$.

We collect the decision variables $\Delta\mathbf{R}_n$ for $n = 1, \ldots, \bar{n}$, in a vector $\boldsymbol{x}$, structured as:

$$\boldsymbol{x}^{\mathsf{T}} = [\boldsymbol{x}_{1,1}^{\mathsf{T}}, \ldots, \boldsymbol{x}_{P,P}^{\mathsf{T}}]$$

with

$$\boldsymbol{x}_{i,j}^{\mathsf{T}} = [(\Delta\mathbf{R}_0)_{i,j}, \ldots, (\Delta\mathbf{R}_{\bar{n}})_{i,j}]$$

where $(\Delta\mathbf{R}_n)_{i,j}$ denotes the (i-th, j-th) entry of the matrix $\Delta\mathbf{R}_n$.

Defining $\beta_{k;n}$ as:

$$\beta_{k;n} = \frac{\varphi_n(\mathrm{j}\omega_k)}{\mathsf{D}(\mathrm{j}\omega_k)}$$

and collecting all the terms, one for each basis function, in a vector $\mathbf{b}_k$:

$$\mathbf{b}_k = [\beta_{k;0}, \ldots, \beta_{k;\bar{n}}]^{\mathsf{T}}$$

it holds that:

$$\mathcal{E}_{i,j}^2 = \|\mathbf{B}_{i,j}\boldsymbol{x}_{i,j}\|_2^2$$

where, $\mathbf{B}_{i,j} = [\mathbf{b}_1^{\mathsf{T}}, \ldots, \mathbf{b}_{\bar{k}}^{\mathsf{T}}]^{\mathsf{T}}$.

As proposed in [18], it is possible to further modify the cost function computing the so called "economy size" QR factorization of the matrix $\mathbf{B}_{i,j}$. In fact $\mathbf{B}_{i,j}$ can be decomposed as:

$$\mathbf{B}_{i,j} = \mathbf{Q}_{i,j}\boldsymbol{\Psi}_{i,j}, \quad \mathbf{Q}_{i,j}^{\mathsf{T}}\mathbf{Q}_{i,j} = \mathbb{I}$$

Since $\mathbf{Q}_{i,j}$ is orthonormal, and recalling that an orthonormal transformation preserves the 2-norm, it holds that:

$$\|\mathbf{B}_{i,j}\boldsymbol{x}_{i,j}\|_2^2 = \|\mathbf{Q}_{i,j}\boldsymbol{\Psi}_{i,j}\boldsymbol{x}_{i,j}\|_2^2 = \|\boldsymbol{\Psi}_{i,j}\boldsymbol{x}_{i,j}\|_2^2$$

Applying this transformation allows to have a numerically more stable solution while reducing computational times.
Finally, the cost-function can be written as:

$$\mathcal{E}^2 = \|\boldsymbol{\Psi}\boldsymbol{x}\|_2^2\,,$$
$$\text{with } \boldsymbol{\Psi} = \text{blkdiag}\{\boldsymbol{\Psi}_{i,j}\}$$

to consider all the possible input/output ports combinations.
It is important to notice that, being the decision variables $\Delta\mathbf{R}_n$ explicit in $\|\boldsymbol{\Psi}\boldsymbol{x}\|_2^2$, the problem in (3.4) has a convex cost-function. If passivity constraints can be written as convex functions of $\Delta\mathbf{R}_n$, the optimization problem in (3.4) reduces to be convex. This means that it always admits an optimal solution and enables the use of well-established solving techniques, such as gradient-descent or interior point methods.

For these reasons, in the next paragraph we will focus on deriving a set of convex passivity constraints for immittance and scattering systems.

Immittance local constraints

In order to derive a set of local passivity constraints for immittance systems, we recall that, to be passive, $\mathbf{H}(s) + \mathbf{H}^{\mathsf{H}}(s)$ must be non-negative definite. Supposing that, for $s_k = \mathrm{j}\omega_k$, the system is not passive, there is for sure at least one eigenvalue of $\mathbf{H}(s_k) + \mathbf{H}^{\mathsf{H}}(s_k)$ smaller than zero. To enforce the system to be passive, it is necessary to "lift" all negative eigenvalues above the zero threshold perturbing the model numerator coefficients through $\Delta\mathbf{H}(s)$.
Unfortunately there is not an explicit relation between the model residues variation and the eigenvalues displacement. However, it is possible to explicitly compute a first order approximation of the eigenvalues variation induced by $\Delta\mathbf{H}(s)$.
To this end, we can define $\Delta\mathbf{H}(s_k) = \Delta\mathbf{H}_k$ as:

$$\Delta\mathbf{H}_k = \frac{\sum_{n=0}^{\bar{n}} \Delta\mathbf{R}_n \varphi_n(s_k)}{\mathsf{D}(s_k)}, \text{ with } \mathsf{D}(s_k) = \sum_{n=0}^{\bar{n}} r_n \varphi_n(s_k)$$

and denote as $\underline{\lambda}_k$ a simple eigenvalue of $\mathbf{H}(s_k) + \mathbf{H}^{\mathsf{H}}(s_k)$.

Applying a perturbation $\Delta\mathbf{H}_k$ to $\mathbf{H}(s_k)$, $\underline{\lambda}_k$ will be correspondingly perturbed, at first order, toward $\widehat{\underline{\lambda}}_k$, as shown in Equation (3.5):

$$\widehat{\underline{\lambda}}_k = \underline{\lambda}_k + \boldsymbol{v}_k^{\mathsf{H}}(\Delta\mathbf{H}_k^{\mathsf{H}} + \Delta\mathbf{H}_k)\boldsymbol{v}_k \tag{3.5}$$

where $\boldsymbol{v}_k$ is the right eigenvector of $\mathbf{H}^{\mathsf{H}}(s_k) + \mathbf{H}(s_k)$, related to $\underline{\lambda}_k$.
In the following, just negative eigenvalues will be considered, to avoid non-active constraints in the final optimization problem.
To derive proper constraints, the decision variables (the numerator coefficients of $\Delta\mathbf{H}_k$), must be de-embedded from (3.5). To this end it is easy to prove that (3.5) can be re-written as:

$$\widehat{\underline{\lambda}}_k = \underline{\lambda}_k + 2 \cdot \mathrm{Re}\left\{\boldsymbol{v}_k^{\mathsf{H}} \Delta\mathbf{H}_k \boldsymbol{v}_k\right\} \tag{3.6}$$

Defining now:

$$\alpha_{k;n} = \frac{\varphi_n(s_k)}{\mathsf{D}(s_k)}$$

it is possible to stack all these terms in a vector:

$$\mathbf{a}_k = [\alpha_{k,0}, \ldots, \alpha_{k,\bar{n}}]^{\mathsf{T}}$$

enabling writing (3.6) as:

$$\widehat{\underline{\lambda}}_k = \underline{\lambda}_k + 2 \cdot \mathrm{Re}\left\{(\boldsymbol{v}_k^{\mathsf{H}} \otimes \boldsymbol{v}_k^{\mathsf{T}}) \otimes \mathbf{a}_k^{\mathsf{T}}\right\} \boldsymbol{x} \tag{3.7}$$

The first order perturbed eigenvalue $\widehat{\underline{\lambda}}_k$ must be set greater than zero to guarantee passivity:

$$\underline{\lambda}_k + 2 \cdot \mathrm{Re}\left\{(\boldsymbol{v}_k^{\mathsf{H}} \otimes \boldsymbol{v}_k^{\mathsf{T}}) \otimes \mathbf{a}_k^{\mathsf{T}}\right\} \boldsymbol{x} \geq 0$$

This relation can be re-arranged as a linear inequality constraint:

$$\mathbf{p}_k^{\mathsf{T}} \cdot \boldsymbol{x} \leq -\underline{\lambda}_k \tag{3.8}$$

with $\mathbf{p}_k^{\mathsf{T}} = -2 \cdot \mathrm{Re}\left\{(\boldsymbol{v}_k^{\mathsf{H}} \otimes \boldsymbol{v}_k^{\mathsf{T}}) \otimes \mathbf{a}_k\right\}$.

Collecting now all the $\bar{c}$ passivity violation locations (frequencies at with the largest violation is attained) in a structure $\mathcal{W}$, it is possible to define a convex polytope, defined by the superposition of linear inequality constraints, that approximates the true feasibility set corresponding to $\mathbf{H}^{\mathsf{H}}(s_k) + \mathbf{H}(s_k) \geq 0$.
Summarizing, the optimization problem in Equation (3.4) can be particularized as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \|\boldsymbol{\Psi}\boldsymbol{x}\|_2^2 \\ &\text{s.t. } \mathbf{P} \cdot \boldsymbol{x} \leq -\boldsymbol{\lambda} \end{aligned} \tag{3.9}$$

with $\mathbf{P} = [\mathbf{p}_1^{\mathsf{T}}, \ldots, \mathbf{p}_{\bar{c}}^{\mathsf{T}}]^{\mathsf{T}}$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{\bar{c}}]^{\mathsf{T}}$.

Scattering local constraints

Local passivity constraints for scattering systems can be formulated both starting from the non-negative definiteness of $\mathbb{I} - \mathbf{H}^{\mathsf{H}}(s)\mathbf{H}(s)$ or from the singular values decomposition of $\mathbf{H}(s)$. Here we follow this second approach.
Denoting as $\mathbf{H}_k = \mathbf{H}(\mathrm{j}\omega_k)$ the model transfer matrix evaluated in $s = \mathrm{j}\omega_k$, it admits the following singular values decomposition:

$$\mathbf{H}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^{\mathsf{H}}$$

where $\mathbf{U}_k, \mathbf{V}_k$ collects, respectively, left and right singular vectors and $\boldsymbol{\Sigma}_k$ has in its main diagonal the singular values of $\mathbf{H}_k$, sorted in descending order.

Recalling the results in Theorem 1.7, a scattering system is passive if, for any frequency, all the singular values are below the unitary threshold. It follows that, to enforce the passivity of a scattering system all the non passive singular values must be enforced to be less than 1 by perturbing numerator residues.
Also in this case, there are no analytic relations linking singular values perturbations induced by a variation on model coefficients. On the other hand, it is possible again to write explicitly the first order approximation of singular values displacement induced by $\Delta\mathbf{H}_k$.
Denoting as $\underline{\sigma}_k$ a singular value of $\mathbf{H}_k$ and perturbing the transfer matrix with $\Delta\mathbf{H}_k$, $\underline{\sigma}_k$ will be correspondingly perturbed, at first order, toward $\widehat{\underline{\sigma}}_k$, as Equation (3.10) shows:

$$\widehat{\underline{\sigma}}_k = \underline{\sigma}_k + \mathrm{Re}\left\{\boldsymbol{u}_k^{\mathsf{H}} \Delta\mathbf{H}_k \boldsymbol{v}_k\right\} \tag{3.10}$$

where $\boldsymbol{u}_k, \boldsymbol{v}_k$ are respectively left and right singular vectors of $\Delta\mathbf{H}_k$ related to $\underline{\sigma}_k$.

Decision variables $\boldsymbol{x}$ are embedded in Equations (3.10). Following the same approach used in the immittance case, it is still possible to de-embed model residues from $\Delta\mathbf{H}_k$. To this end, keeping the same ordering as in Equation (3.7), it is possible to write Equation (3.10) as:

$$\widehat{\underline{\sigma}}_k = \underline{\sigma}_k + \mathrm{Re}\left\{(\boldsymbol{v}_k^{\mathsf{H}} \otimes \boldsymbol{u}_k^{\mathsf{T}}) \otimes \mathbf{a}_k^{\mathsf{T}}\right\} \cdot \boldsymbol{x} \tag{3.11}$$

where it still holds that $\mathbf{a}_k = \left[\frac{\varphi_0(s_k)}{\mathsf{D}(s_k)}, \ldots, \frac{\varphi_{\bar{n}}(s_k)}{\mathsf{D}(s_k)}\right]^{\mathsf{T}}$.

Imposing scattering systems passivity constraint ($\widehat{\underline{\sigma}}_k \leq 1$) in Equation (3.11), it turns out that:

$$\underline{\sigma}_k + \mathrm{Re}\left\{(\boldsymbol{v}_k^{\mathsf{H}} \otimes \boldsymbol{u}_k^{\mathsf{T}}) \otimes \mathbf{a}_k^{\mathsf{T}}\right\} \cdot \boldsymbol{x} \leq 1$$

Re-arranging the terms, we get to the linear inequality constraint:

$$\mathbf{f}_k^{\mathsf{T}} \cdot \boldsymbol{x} \leq 1 - \underline{\sigma}_k \tag{3.12}$$

with $\mathbf{f}_k^{\mathsf{T}} = \mathrm{Re}\left\{(\boldsymbol{v}_k^{\mathsf{H}} \otimes \boldsymbol{u}_k^{\mathsf{T}}) \otimes \mathbf{a}_k^{\mathsf{T}}\right\}$.

Collecting now all the $\bar{c}$ passivity violation locations (frequencies at which the largest violation is attained) in a structure $\mathcal{W}$, the true feasibility set, defined by $\mathbb{I} - \mathbf{H}^{\mathsf{H}}(s)\mathbf{H}(s) \geq 0$, can be approximated through a set of linear inequality constraints by a convex polytope. The general optimization problem in Equation (3.4), can be particularized for scattering systems as:

$$\begin{aligned} &\min_{\boldsymbol{x}} \|\boldsymbol{\Psi}\boldsymbol{x}\|_2^2 \\ &\text{s.t. } \mathbf{F} \cdot \boldsymbol{x} \leq \mathbf{s} \end{aligned} \tag{3.13}$$

with $\mathbf{F} = [\mathbf{f}_1^{\mathsf{T}}, \ldots, \mathbf{f}_{\bar{c}}^{\mathsf{T}}]^{\mathsf{T}}$ and $\mathbf{s} = [1 - \underline{\sigma}_1, \ldots, 1 - \underline{\sigma}_{\bar{c}}]^{\mathsf{T}}$.

3.3 Passivity Assessment

In this section, we address the problem of how to properly find and collect the location of passivity locations. The main issue in getting local passivity constraints, is where to search for passivity violations along the frequency dimension, without losing any critical point.

Almost every frequency-sampling based technique is going to fail. In fact, recalling Theorems 1.2 and 1.5, the entire frequency axis $\omega \in [0, \infty]$ must be checked, and potentially an infinite number of sampling points have to be processed. A restriction of this set to the system bandwidth is not allowed because, when the macro-model is used in simulation, the embedded simulation engine may excite frequencies outside the system bandwidth that, if not passive, may lead to a general failure due to unstable solutions.

The Hamiltonian-driven method is a purely algebraic passivity assessment scheme that allows to accurately pinpoint violations along the frequency axis thus avoiding brute-force sampling methods. Hamiltonian matrices related to system realizations, recalling what already stated in Section 1.6, can be used to algebraically find frequency values at which passivity violations occurs. The main result of Section 1.6 is that purely imaginary eigenvalues of a Hamiltonian matrix (or pencil), denoted as λ_{im}, associated with the model, equal exactly the complex frequencies at which passivity violations occur.
Imaginary Hamiltonian eigenvalues induce then a partitioning on the frequency axis in passive and non-passive bands. To decide whether a band is passive or not, it is sufficient to check if there exist negative eigenvalues of $\mathbf{H}(\mathrm{j}\omega) + \mathbf{H}^{\mathsf{H}}(\mathrm{j}\omega)$, if in immittance representation, or singular values of $\mathbf{H}(\mathrm{j}\omega)$ that exceed one, if in scattering representation.
Focusing now our interest just in non-passive bands, we need to find the largest passivity violation (for each eigenvalue/singular value) inside this band and formulate proper local constraints. To this end, each non-passive band is finely sampled looking for the minimum eigenvalue/maximum singular value, in order to reduce the computational effort during the passivity enforcement.
This procedure to find non-passive areas is used inside a passivity verification algorithm that, storing violation locations, is the core of passivity enforcement algorithms.
A sketch of the algorithm used to localize passivity violations is shown in Algorithm 3.1.

Algorithm 3.1 Passivity test algorithm

Require: Non-passive model
1: Define violation structure $\mathcal{W}$ as empty
2: Build Hamiltonian matrix $\boldsymbol{\mathcal{M}}$ or pencil $(\boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{K}})$ as in Section 1.6
3: Solve the corresponding eigen-problem to get Hamiltonian imaginary eigenvalues λ_{im}
4: **if** $\lambda_{im} \neq \emptyset$ **then**
5: Partition the frequency axis in $\mathbf{card}(\lambda_{im}) + 1$ sub-bands
6: Define μ non-passive sub-bands
7: **for** $b = 1, \ldots, \mu$ **do**
8: Sample the b-th band to find the largest violation for each eigenvalue/singular value
9: Stack the new violation in $\mathcal{W}$
10: **end for**
11: **end if**
12: **return** Passivity violations $\mathcal{W}$

3.4 Passivity Enforcement

In Section 3.2 we derived a general scheme to impose passivity on a non-passive model through a convex optimization problem, subject to linear inequality constraints for both immittance and scattering representations. In Section 3.3 we discussed an algebraic method to find where, along the frequency axis, formulate these constraints.
Now we are ready to properly state a passivity enforcement scheme (for further details

see [21]).

Before proceeding in the algorithm definition, some remarks about our theoretical framework are in order. The main assumption underlying the formulation of passivity constraints in Section 3.2.2, is that the eigenvalues/singular values perturbations induced by model coefficients variations are linear. However, approximating the true feasibility set with a polytope may lead to inaccuracies and, even if the linearized problems admits feasible solutions, these in general do not coincide with the ones of (3.4).
On the other hand, even if these linearized constraints differ from the true feasibility set, it has been noted that they lead to a reduction of violation areas when adopted in model passivity enforcement. Thus, even though the solutions of optimization problems (3.9) and (3.13) do not return the passive model residues, this result enables the use of passivity enforcement iterative algorithms where, as the number of iterations grows, model passivity is enforced uniformly throughout $\omega \in \mathbb{R}$.
This approach is now a common practice in solving passivity enforcement problems, since the use of full non-linear constraints requires an LMI optimization (Lemma 1.1 and 1.2), whose computational cost is not manageable when the number of ports or the model dynamical order increases [21].

The proposed iterative passivity enforcement algorithm relies on Algorithm 3.1 to get passivity violation locations and solves iteratively convex optimization problems in (3.9) and (3.13) to reduce, iteration after iteration, passivity violations, until a passive model is obtained. This passivity enforcement scheme is reported in Algorithm 3.2.

Algorithm 3.2 Passivity enforcement algorithm

Require: Model coefficients $\mathbf{R}_n,\ r_n$ for $n = 1, \ldots, \bar{n}$
1: Get passivity violations structure $\mathcal{W}$ computed as in Algorithm 3.1
2: **while** $\mathcal{W}$ not empty **do**
3: Build constraints as in (3.8) or (3.12)
4: Build cost function regressor $\mathbf{\Psi}$
5: Get residues perturbations $\Delta\mathbf{R}_n$ solving convex optimization problem as in (3.9) or (3.13)
6: Update model coefficients as $\mathbf{R}_n \leftarrow \mathbf{R}_n + \Delta\mathbf{R}_n$
7: Get passivity violations vector $\mathcal{W}$ of the perturbed model with algorithm 3.1
8: **end while**
9: **return** Passive model coefficients

While the linear approximation on constraints allows to solve in reasonable time even high-order problems, it has a major counter-effect: there is no guarantee that the algorithm converges to a passive model.
In fact, it is possible that it enters in a "cycle" of period γ where, after γ passivity enforcement iterations, the perturbed model is identical to the first one, never converging to a passive model. This undesired phenomenon is strictly related to the constraint linearization. Fortunately, robust formulations that are guaranteed to converge are known. One of these relies on a "constraint prediction" scheme, that is able to avoid oscillating

behaviours. The main idea is to stack in a single constraints vector the ones obtained from a given number of successive passivity enforcement iterations δ, in such a way that all the intermediate constraints are used in a final optimization problem. Geometrically, stacking δ constraint sets all together means that we are locally reducing the error with which the linear constraints polytope approximates the true (non-linear) feasibility set, thus reducing the error induced by the constraints linear approximation.
In details, denoting as $\overline{\mathbf{H}}(s)$ a non passive model at the k-th passivity enforcement iteration, we collect, through an inner enforcement loop, additional constraints, to be used in an outer optimization problem. In this inner loop the enforcement Algorithm 3.2, starting from $\overline{\mathbf{H}}(s)$, works on intermediate models $\underline{\mathbf{H}}^{(i)}(s)$, where i denotes the inner loop iteration index, that are used to collect sets of violations locations $\mathcal{W}_i$ related to a number δ of successive enforcement iterations. The main difference of this robust implementation with respect to the previous one is that, instead of deleting information about constraints at previous iterations, it keeps them in memory.
Once a suitable set of δ successive constraints set has been stored, a final passivity enforcement is performed on the initial model $\overline{\mathbf{H}}(s)$, under the extended set of constraints defined by the violations $\mathcal{W}_{ext} = [\mathcal{W}_1^\mathsf{T}, \ldots, \mathcal{W}_\delta^\mathsf{T}]^\mathsf{T}$. The resulting perturbed model, if still not passive, will be used in the next robust passivity enforcement iteration.
It may happen that, if no more oscillating solutions arise, the inner enforcement loop converges to a passive model. In this case the inner loop returns a number $0 < \delta^* < \delta$ of stacked set of constraints, that are directly cast in the final optimization problem.

Indicating respectively with $\overline{\mathbf{R}}_n$, $\underline{\mathbf{R}}_n^{(i)}$ the n-th numerator residue of $\overline{\mathbf{H}}(s)$, $\underline{\mathbf{H}}^{(i)}(s)$ and with $\Delta\overline{\mathbf{R}}_n$, $\Delta\underline{\mathbf{R}}_n^{(i)}$ the outer and inner loop perturbation coefficients, this passivity enforcement robust implementation is described in Algorithm 3.3

Numerical results show that in most of the cases the non-robust implementation is enough to guarantee the algorithm convergence to a passive model, keeping the responses accurate. However, in all the cases where oscillatory solutions arise, the robust implementation might be used. The computational effort in this case, however, is possibly higher.

Algorithm 3.3 Robust passivity enforcement algorithm

Require: Non-passive model ($\overline{\mathbf{H}}(s)$) coefficients $\overline{\mathbf{R}}_n$ for $n = 1, \ldots, \bar{n}$
Require: Number of prediction iterations δ
Get current model passivity violations vector $\mathcal{W}$ as in Algorithm 3.1
while $\mathcal{W} \neq \emptyset$ **do**
    Set $\underline{\mathbf{H}}^{(0)}(s) = \overline{\mathbf{H}}(s)$ and $\mathcal{W}_1 = \mathcal{W}$
    Set $i = 1$ and $\mathcal{W}_{ext} = \emptyset$
    while $i \leq \delta$ **do**
        Get perturbation coefficients $\Delta\underline{\mathbf{R}}_n^{(i)}$ as in Algorithm 3.2, given $\mathcal{W}_i$
        Update $\underline{\mathbf{H}}^{(i)}(s)$ coefficients as $\underline{\mathbf{R}}_n^{(i+1)} \leftarrow \underline{\mathbf{R}}_n^{(i)} + \Delta\underline{\mathbf{R}}_n^{(i)}$ to get the perturbed intermediate model $\underline{\mathbf{H}}^{(i+1)}(s)$
        Get $\underline{\mathbf{H}}^{(i+1)}(s)$ passivity violations $\mathcal{W}_{i+1}$ as in Algorithm 3.1
        if $\mathcal{W}_{i+1} \neq \emptyset$ **then**
            Stack passivity violations vector $\mathcal{W}_{i+1}$ in $\mathcal{W}_{ext} = [\mathcal{W}_{ext}^{\mathsf{T}},\ \mathcal{W}_{i+1}^{\mathsf{T}}]^{\mathsf{T}}$
        else
            break
        end if
        $i \leftarrow i + 1$
    end while
    Get perturbation coefficients $\Delta\overline{\mathbf{R}}_n$ as in Algorithm 3.2 with the extended set of passivity violations $\mathcal{W}_{ext}$
    Update $\overline{\mathbf{H}}(s)$ coefficients as $\overline{\mathbf{R}}_n \leftarrow \overline{\mathbf{R}}_n + \Delta\overline{\mathbf{R}}_n$
    Get passivity violations $\mathcal{W}$ of the perturbed model
end while
return Passive model coefficients

Chapter 4

Passivity Enforcement Schemes for Mono-variate Macromodels

The extraction of passive macro-models becomes of major importance when parametric simulations are required since, with the help of reduced-order models, computational times can be reduced of several orders of magnitude with respect to classical techniques that, often, involve the use of full-wave electromagnetic solvers. As shown in Chapter 2, the well-established Parametric Sanathanan-Koerner iteration (PSK) ([10,33,35,37]) allows to extract parametric macro-models from sets tabulated data. However, as for the non-parameterized case, this estimation algorithm does not embed any passivity constraint, so that the synthesized models are not guaranteed passive neither along the frequency axis nor along the parameters ones. This leads to the necessity of parametric passivity enforcement schemes that are able to enforce passivity uniformly along both the frequency and parameters dimensions. In this Chapter we will discuss a known passivity verification and enforcement algorithm for parameterized macro-models, in view of the further developments that will be presented in Chapters 5 and 6.

4.1 Passivity on a Bi-Dimensional Space

As suggested in the introductory part of this Chapter, dealing with parametric macro-models, whose structure is recalled in Equation (4.1):

$$\mathbf{H}(s;\vartheta) = \frac{\mathbf{N}(s,\vartheta)}{\mathsf{D}(s,\vartheta)} = \frac{\sum_{n=0}^{\bar{n}} \sum_{\ell=1}^{\bar{\ell}} \mathbf{R}_{n,\ell}\, \xi_\ell(\vartheta)\, \varphi_n(s)}{\sum_{n=0}^{\bar{n}} \sum_{\ell=1}^{\bar{\ell}} r_{n,\ell}\, \xi_\ell(\vartheta)\, \varphi_n(s)} \tag{4.1}$$

implies an extension to a multi-dimensional space of the passivity verification method detailed in Chapter 3. In fact, if in the non-parametric case the passivity must be verified just along one dimension, for parametric models we must check for violations in a $(\rho+1)$-dimensional space, with ρ the number of external parameters.
In this Chapter we consider just the case in which $\rho = 1$, thus the space in which we search for violations is a subset of $\mathbb{R}^2$.

As we have extensively shown in Sections 3.2–3.4, the Hamiltonian-driven scheme is the main tool that is used to spot passivity violations along the frequency axis. In principle, to be sure that the model is uniformly passive for any feasible parameter value, it would be necessary to perform non-parameterized Hamiltonian tests for a large set of parameter values. Unfortunately, eigenvalues extraction, that is the main operation in the Hamiltonian verification method, is a computationally expensive task, especially when the matrix dimension increases, as in the case of parametric macro-models. Thus, a parametric passivity assessment method that pursues this line, apart from being useless in some cases, would lead to un-manageable computational times. For these reasons it has been chosen to develop an adaptive algorithm that, according to some rules that will be detailed later on, samples the parameter space just where it is required, without the intervention of the user.

In the next section we will introduce some aspects of parameterized Hamiltonian eigenvalues, that will be extensively used throughout this work.

4.2 Parametrized Hamiltonian Eigenvalues

The Hamiltonian matrix associated with the system contains the model realization matrices, that are parameter dependent as shown in Section 2.1.2. This means that a parameter variation induces a variation on the Hamiltonian matrix coefficients, thus on its eigenvalues and eigenvectors. Figure 4.1 shows Hamiltonian eigenvalues for a parametric macro-model as a function of the parameter ϑ.
Recalling the results in Theorems 1.4 and 1.7 we know that, in order to make a precise passivity characterization for a parametric model, our aim is to find all the parameter values for which the Hamiltonian matrix has, at least, a pair of purely imaginary eigenvalues. Unfortunately, the relation that links these eigenvalues with the parameter is too complex and can not be used to predict if, for a given parameter value, the system is passive or not.
However, even though an algebraic parametric passivity verification scheme is unavailable, we can exploit the Hamiltonian eigenvalues implicit parameterization to adaptively search along the parameter axis to detect where violation areas are likely to be found.
To this end, we should sample the parameter space just in these areas where Hamiltonian eigenvalues trajectories are close to the imaginary axis. In these cases, in fact, under a small parameter variation some eigenvalues may become purely imaginary, leading to a passivity violation.
In the next section we will detail how Hamiltonian eigenvalues trajectories can be used to build a parametric passivity assessment algorithm.

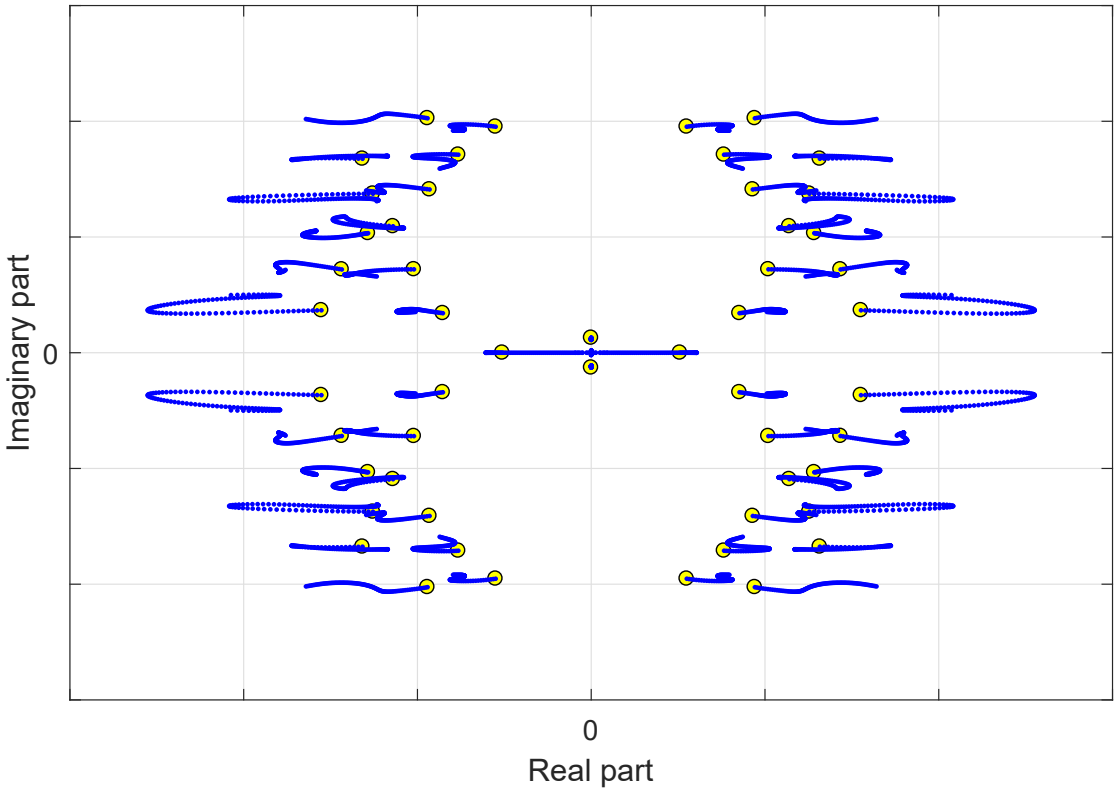


Figure 4.1: Parameterized Hamiltonian eigenvalues

4.3 Parametrized Adaptive Passivity Verification Scheme

4.3.1 Adaptive Sampling Strategy

In this section we present a parametric passivity verification and enforcement method, presented originally in [18]. Previously, we focused our attention on the implicit parameterization of Hamiltonian eigenvalues. Moreover, we stated that, to detect violation areas, the main metric we use is the distance of these eigenvalues with respect to the imaginary axis. In order to measure this distance, we introduce a scalar function of the parameter $\Psi(\vartheta)$, defined as:

$$\Psi(\vartheta) = \min_{\lambda(\vartheta)\in\Lambda(\vartheta)} \frac{|\mathrm{Re}\{\lambda(\vartheta)\}|}{\rho(\vartheta)} \tag{4.2}$$

where $\Lambda(\vartheta)$ denotes the Hamiltonian eigen-spectrum evaluated in ϑ and $\rho(\vartheta)$ is the spectral radius, defined as:

$$\rho(\vartheta) = \max_{\lambda(\vartheta)\in\Lambda(\vartheta)} |\lambda(\vartheta)|$$

This function, denoted as "*spectral distance*", returns the real part absolute value of the closest eigenvalue with respect to the imaginary axis and provides a way to check whether

a parameter value is critical or not. In fact, we detect a passivity violation when there are purely imaginary Hamiltonian eigenvalues that, by definition, make $\Psi(\vartheta)$ to vanish throughout the violation area. Thus, in this framework, the main idea to find violations is to adaptively reconstruct $\Psi(\vartheta)$ from a limited set of data, retrieved from an initial coarse linear sampling, to detect where it is likely to vanish. In details, the initial sampling partitions the parameter space in $\bar{\mu}$ sub-intervals whose edges are:

$$\theta_\mu = \vartheta_{min} + \frac{\mu}{\bar{\mu}}(\vartheta_{max} - \vartheta_{min}),\ \mu = 1, ..., \bar{\mu}$$

Knowing the values of $\Psi(\vartheta)$ at these points enables to check if other samples are necessary to better approximate it. If the function is sufficiently smooth, no additional samples are required while, if $\Psi(\vartheta)$ is particularly ill-behaved, we need to add samples.
A fundamental characteristic of this function is that, in general, it is not uniformly differentiable throughout the parameter space. In fact, recalling its definition, if one eigenvalue is moving away from the imaginary axis and, meanwhile, another one is moving in opposite direction, there will be a non-differentiability point when the real parts of the two eigenvalues coincide. Moreover, the function is not differentiable when violations occur.
The non-differentiability plays its role in the process of reconstructing $\Psi(\vartheta)$. In fact, to properly approximate an unknown function from a set of points, we must be aware that we can not use approximating functions that are smoother than the one we are trying to approximate. Being $\Psi(\vartheta) \in C^0$, this consideration leads us to use at most piecewise approximating functions.

As mentioned before, a subinterval defined by its edges $\{\theta_\mu, \theta_{\mu+1}\}$ needs additional samples (from now on we denote this process as "*refining*") if $\Psi(\vartheta)$ does not satisfy some smoothness condition for $\vartheta \in \{\theta_\mu, \theta_{\mu+1}\}$. In order to measure the smoothness of $\Psi(\vartheta)$ in this sub-interval, recalling that just linear functions are allowed, we use as metric a "*linear interpolation error*".
A linear interpolation error approach means that, to decide if the sub-interval must be refined, we measure the distance between the "true" $\Psi(\vartheta)$ and the linear approximating line, for a given parameter point $\theta^* \in [\theta_\mu, \theta_{\mu+1}]$. Assuming that the maximum deviation between the two is attained for:

$$\theta^* = \theta_{\mu+\frac{1}{2}} = \frac{1}{2}(\theta_\mu + \theta_{\mu+1})$$

the linear interpolation error, denoted as $\epsilon_{\mu+\frac{1}{2}}$, reads:

$$\epsilon_{\mu+\frac{1}{2}} = \left| \Psi(\theta_{\mu+\frac{1}{2}}) - \frac{1}{2}[\Psi(\theta_\mu) + \Psi(\theta_{\mu+1})] \right| \tag{4.3}$$

If $\epsilon_{\mu+\frac{1}{2}} > \gamma$, with γ a given threshold, then the function can not be approximated with sufficient precision (dependent on the value of γ) and we need additional samples inside $\{\theta_\mu, \theta_{\mu+1}\}$. Figure 4.2 shows graphically this approach.

In addition to that, in order to better characterize violation areas, some other information

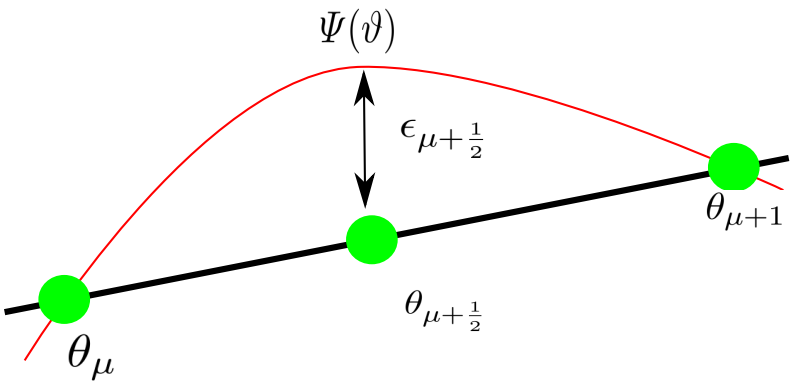


Figure 4.2: Linear interpolation error representation

about the Hamiltonian spectrum can be exploited for setting up the adaptive strategy. Thus, denoting as $\nu(\vartheta)$ the number of purely imaginary Hamiltonian eigenvalues for a fixed parameter ϑ, we can identify four sub-cases that may occur:

1. If $\Psi(\theta_\mu) = \Psi(\theta_{\mu+1}) = 0$ and $\nu(\theta_\mu) = \nu(\theta_{\mu+1})$ we can infer that we spotted a uniformly non passive area, so no refinement to the interval $[\theta_\mu,\ \theta_{\mu+1}]$ is necessary.

2. If $\Psi(\theta_\mu) = \Psi(\theta_{\mu+1}) = 0$ and $\nu(\theta_\mu) \neq \nu(\theta_{\mu+1})$ we spotted a non passive area in which the number of non passive bands changes from θ_μ to $\theta_{\mu+1}$. Thus a refinement must be performed to obtain a precise characterization.

3. If $\Psi(\theta_\mu) > 0$ and $\Psi(\theta_{\mu+1}) = 0$ or, conversely, $\Psi(\theta_\mu) = 0$ and $\Psi(\theta_{\mu+1}) > 0$, a non passive region is respectively opening and closing, then it is necessary to refine in order to spot the exact point in which the violation occurs.

4. If $\Psi(\theta_\mu) > 0$ and $\Psi(\theta_{\mu+1}) > 0$ we can be both in a uniformly passive region, or a non passive region may open and close inside our sampling interval. Thus, in order to discriminate between these two cases, we check if $\Psi(\vartheta)$ present critical behaviours inside the subinterval through the linear interpolation error, defined in (4.3). If this error exceeds a threshold γ the subinterval must be refined.

In order to finely reconstruct $\Psi(\vartheta)$, this refinement process is cast in a iterative loop that, at each iteration, samples, if required, additional points $\theta_{\mu+\frac{1}{2}}$ and adds them to the previous subset. This process stops when $\Psi(\vartheta)$ is approximated satisfactorily by the set of sampled points or a maximum number of iterations M is reached.

4.3.2 Adaptive Sampling Algorithm

In the previous section we detailed how to adaptively sample the parameter space to find passivity violations. In the following, we will discuss in details how these violations are characterized in a 2-dimensional space. In fact, in contrast with the non-parametric case, in order to formulate a single local constraint, we must identify a tuple of frequency-parameter values corresponding to the largest violation. To this end, we know that,

for any feasible parameter value θ_μ, Hamiltonian imaginary eigenvalues $\omega_i(\theta_\mu)$ induce a partition in the frequency axis in $\nu(\theta_\mu)+1$ sub-bands:

$$\Omega_i(\theta_\mu) = \{\omega_i(\theta_\mu),\ \omega_{i+1}(\theta_\mu)\},\quad i = 0,\ldots,\nu(\theta_\mu)$$

that can be locally passive or not passive. In order to include the DC point and to consider asymptotic passivity violations we set $\omega_0 = 0,\ \omega_{\nu(\theta_\mu)+1} = +\infty$.
The worst case passivity violation, attained for $\vartheta = \theta_\mu$, in the i-th non-passive band, is identified by the minimum eigenvalue $\underline{\lambda}_{i,\mu}$ of $\mathbf{H}(\mathrm{j}\bar{\omega}_i;\theta_\mu) + \mathbf{H}^{\mathsf{H}}(\mathrm{j}\bar{\omega}_i;\theta_\mu)$ for immittance systems or by the maximum singular value $\underline{\sigma}_{i,\mu}$ of $\mathbf{H}(\mathrm{j}\bar{\omega}_i;\theta_\mu)$ for scattering systems. In the following we will denote this location in the frequency-parameter space as $(\bar{\omega}_i,\theta_\mu)$. At the end of the passivity check, these violations are stored in a data-structure $\mathcal{W}$ as triplets $(\bar{\omega}_i,\theta_\mu,\underline{\lambda}_{i,\mu})$ for immittance systems or $(\overline{\omega}_i,\theta_\mu,\underline{\sigma}_{i,\mu})$ for scattering ones.

A final remark is in order about the number of initial samples. In fact, even if the initial coarse sampling is followed by, at most, M refined passes, the number of initial point is critical: if is too low we may lose some important violation areas while, if it is too high, it may take an excessively long time to be accomplished. The amount of initial samples is directly related with the model variation induced by the parameter. Here we assume, as detailed in Section 2.1.1, that the parameter basis functions are smooth throughout the parameter space. Thus, due to this property we can use the heuristic rule in (4.4) to find the number of initial sub-intervals.

$$\bar{\mu}_0 = \kappa\bar{\ell},\ \text{with } \kappa > 1 \tag{4.4}$$

The complete adaptive refinement strategy is described in Algorithm 4.4

4.3.3 Numerical Results

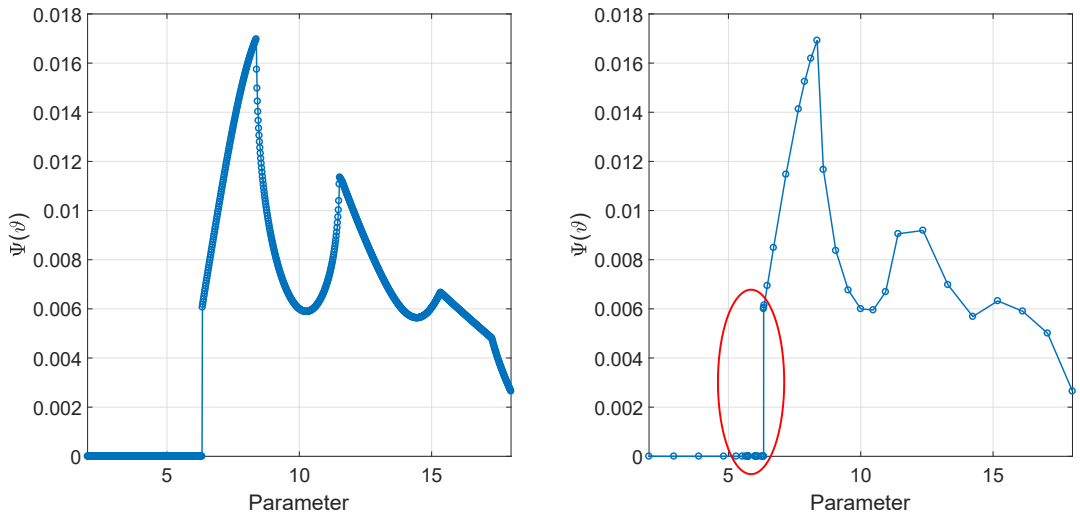
In the following we present some numerical results. We will mainly focus on the ability of the algorithm to adaptively reconstruct $\Psi(\vartheta)$ by comparing its outcomes with a brute-force sampling on the parameter space. Figures 4.3a and 4.3b make this comparison on a real test-case (*Test Case 2–a*, see Appendix A).
We can see from Figure 4.3b that, even with a smaller number of points, the adaptive strategy is able to approximate $\Psi(\vartheta)$ as satisfactorily as the brute-force sampling does, whose result is shown in Figure 4.3a. Moreover, we see that the adaptive algorithm refines, as detailed above, just in critical areas, highlighted in red in Figure 4.3b. This allows us to be sure, at a level given by the threshold γ, that no other violations are present in the parameter space, leading to a considerable time saving. In fact, the brute-force sampling needs 23.6 s to be accomplished (10^3 points), while the adaptive one just 6.6 s being, more or less, 4 times faster but still reliable.

4.4 Parametric Passivity Enforcement

Once the parametric passivity check is performed, if the model is not passive, we must enforce its passivity through a parametric passivity enforcement [18]. The procedure we

Algorithm 4.4 Parametric passivity verification

Require: Parameter basis order $\bar{\ell}$
Require: Model state-space/descriptor realization
Require: Control parameters M, κ
1: Define the number of initial sub-intervals μ_0 as in (4.4)
2: Sample initial parameter points and store them in $\mathcal{S}_0 = \{\theta_\mu,\ \mu = 0, \dots, \bar{\mu}_0\}$
3: Set $m = 0$
4: **while** $\mathcal{S}_m$ is not empty or $m \leq M$ **do**
5: **for** $\mu = 1, \dots, \bar{\mu}_m$ **do**
6: Build Hamiltonian matrix $\mathbf{M}(\theta_\mu)$ or the Skew Hamiltonian/Hamiltonian pencil $(\mathbf{M}(\theta_\mu), \mathbf{K})$ as detailed in Section 1.6
7: Extract Hamiltonian imaginary eigenvalues $\omega_i(\theta_\mu)$, with $i = 1, \dots, \nu(\theta_\mu)$
8: Extract local worst case violations $(\bar{\omega}_i, \theta_\mu, \underline{\sigma}_{i,\mu})$ or $(\bar{\omega}_i, \theta_\mu, \underline{\lambda}_{i,\mu})$ for each non-passive band and store them in $\mathcal{W}$
9: **end for**
10: $m \leftarrow m + 1$
11: Determine new samples $\mathcal{S}_m$ according to the adaptive strategy
12: Redefine the number of parameter space sub-intervals $\bar{\mu}_m$ to be checked at next iteration
13: **end while**
14: **return** Passivity violations locations $\mathcal{W}$



(a) Brute force $\Psi(\vartheta)$ sampling

(b) Adaptive $\Psi(\vartheta)$ sampling

Figure 4.3: Brute force/Adaptive sampling strategies comparison (*Test Case 2–a*)

pursue to enforce the model passivity is based again on numerator residues perturbations, as in the non-parameterized case. Defining with $\hat{\mathbf{H}}(s;\vartheta)$ the passive model and with $\mathbf{H}(s;\vartheta)$ the not passive one, we want to find the numerator residues of a transfer matrix

$\Delta\mathbf{H}(s;\vartheta)$ that allow to bring $\mathbf{H}(s;\vartheta)$ toward $\hat{\mathbf{H}}(s;\vartheta)$. The perturbation matrix is defined as:

$$\Delta\mathbf{H}(s;\vartheta) = \frac{\Delta\mathbf{N}(s,\vartheta)}{\mathsf{D}(s,\vartheta)} = \frac{\sum_{n=0}^{\bar{n}}\sum_{\ell=1}^{\bar{\ell}}\Delta\mathbf{R}_{n,\ell}\,\xi_\ell(\vartheta)\,\varphi_n(s)}{\sum_{n=0}^{\bar{n}}\sum_{\ell=1}^{\bar{\ell}} r_{n,\ell}\,\xi_\ell(\vartheta)\,\varphi_n(s)}, \tag{4.5}$$

The perturbation residues are computed through the optimization problem in (4.6):

$$\begin{aligned} &\min \|\Delta\mathbf{H}(s;\vartheta)\|^2 \\ &\text{s.t. } \mathbf{H}(s;\vartheta) + \Delta\mathbf{H}(s;\vartheta) \text{ is passive.} \end{aligned} \tag{4.6}$$

Before going into the details on how to derive parametric passivity constraints, some algebraic manipulation on the cost-function are required to explicit the decision variables. Following the same scheme, as in Section 3.2.2, we rewrite the cost function as:

$$\mathcal{E}^2 = \sum_{i,j=1}^{P}\mathcal{E}_{i,j}^2, \qquad \mathcal{E}_{i,j}^2 = \sum_{k=1}^{\bar{k}}\sum_{m=1}^{\bar{m}}|\Delta H_{i,j}(\mathrm{j}\omega_k;\vartheta_m)|^2 \tag{4.7}$$

where k and m are, respectively, frequency and parameter sample indices, and (i,j) denote the (i-th, j-th) element of the matrix $\Delta\mathbf{H}(\mathrm{j}\omega_k;\vartheta_m)$.

Assuming that the decision variables are organized in a vector $\boldsymbol{x}$ as:

$$\boldsymbol{x} = [\boldsymbol{x}_{1,1}^{\mathsf{T}},\ldots,\boldsymbol{x}_{i,j}^{\mathsf{T}},\ldots,\boldsymbol{x}_{P,P}^{\mathsf{T}}]^{\mathsf{T}}$$

where

$$\boldsymbol{x}_{i,j} = [(\Delta\mathbf{R}_{0,1})_{i,j},\ldots,(\Delta\mathbf{R}_{n,\ell})_{i,j},\ldots,(\Delta\mathbf{R}_{\bar{n},\bar{\ell}})_{i,j}]^{\mathsf{T}}$$

We can define $\beta_{k,m;n,\ell}$ as:

$$\beta_{k,m;n,\ell} = \frac{\xi_\ell(\vartheta_m)\varphi(\mathrm{j}\omega_k)}{\mathsf{D}(\mathrm{j}\omega_k,\vartheta_m)}$$

and

$$\mathbf{b}_{k,m}^{\mathsf{T}} = [\beta_{k,m;0,1},\ldots,\beta_{k,m;n,\ell},\ldots,\beta_{k,m;\bar{n},\bar{\ell}}]$$

thus, it holds that:

$$\mathcal{E}_{i,j}^2 = \|\mathbf{B}_{i,j}\;\boldsymbol{x}_{i,j}\|_2^2 \tag{4.8}$$

with $\mathbf{B}_{i,j} = [\mathbf{b}_{1,1}^{\mathsf{T}},\ldots,\mathbf{b}_{\bar{k},\bar{m}}^{\mathsf{T}}]^{\mathsf{T}}$.

Recalling the properties of the QR-factorization applied to a least-square problem (see Section 3.2.2), Equation (4.8) can be written as:

$$\mathcal{E}_{i,j}^2 = \|\boldsymbol{\Xi}_{i,j}\boldsymbol{x}_{i,j}\|_2^2 \tag{4.9}$$

where $\mathbf{B}_{i,j} = \mathbf{Q}_{i,j}\;\boldsymbol{\Xi}_{\mathbf{i,j}}$ and $\mathbf{Q}_{i,j}^{\mathsf{T}}\;\mathbf{Q}_{i,j} = \mathbb{I}$
Thus, the cost function in 4.6 can be cast in the following form:

$$\begin{aligned} &\min \|\boldsymbol{\Xi}\;\boldsymbol{x}\|_2^2\,, \\ &\boldsymbol{\Xi} = \mathrm{blkdiag}\{\boldsymbol{\Xi}_{i,j}\}_{i,j=1}^{P} \end{aligned}$$

To formulate parametric passivity constraints we will pursue again a local approach. In the following, considering a local worst case violation occurring at $\omega = \overline{\omega}_i$ and $\vartheta = \theta_\mu$, we will derive as in Section 3.2.2 linearized local passivity constraints for immittance and scattering systems.

4.4.1 Parametric Immittance Systems

Denoting as $\underline{\lambda}_{i,\mu}$ the smallest eigenvalue of $\mathbf{H}^{\mathsf{H}}(\mathrm{j}\bar{\omega}_i;\theta_\mu)+\mathbf{H}(\mathrm{j}\bar{\omega}_i;\theta_\mu)$, under a perturbation matrix $\Delta\mathbf{H}(\mathrm{j}\bar{\omega}_i;\theta_\mu)$, $\underline{\lambda}_{i,\mu}$ will be correspondingly perturbed, at first order, toward $\widehat{\lambda}_{i,\mu}$ as shown in (4.10):

$$\widehat{\lambda}_{i,\mu} = \underline{\lambda}_{i,\mu} + \mathrm{Re}\left\{\boldsymbol{v}_{i,\mu}^{\mathsf{H}}\Delta\mathbf{H}(\mathrm{j}\bar{\omega}_{i,\mu};\theta_\mu)\boldsymbol{v}_{i,\mu}\right\} \tag{4.10}$$

where $\boldsymbol{v}_{i,\mu}$ is the right eigenvector of $\mathbf{H}^{\mathsf{H}}(\mathrm{j}\bar{\omega}_i;\theta_\mu) + \mathbf{H}(\mathrm{j}\bar{\omega}_i;\theta_\mu)$ associated with $\underline{\lambda}_{i,\mu}$. For the derivation see 3.2.2.

Thus, defining $\alpha_{i,\mu;n,\ell}$ as:

$$\alpha_{i,\mu;n,\ell} = \frac{\xi_\ell(\theta_\mu)\varphi_n(j\bar{\omega}_{i,\mu})}{\mathsf{D}(j\bar{\omega}_{i,\mu},\theta_\mu)}$$

and

$$\mathbf{a}_{i,\mu} = [\alpha_{i,\mu;0,1},\ldots,\alpha_{i,\mu;n,\ell},\ldots,\alpha_{i,\mu;\bar{n},\bar{\ell}}]^{\mathsf{T}}$$

Equation (4.10) can be cast as:

$$\widehat{\lambda}_{i,\mu} = \underline{\lambda}_{i,\mu} + \mathrm{Re}\left\{(\boldsymbol{v}_{i,\mu}^{\mathsf{H}}\otimes\boldsymbol{v}_{i,\mu}^{\mathsf{T}})\otimes\mathbf{a}_{i,\mu}^{\mathsf{T}}\right\}\boldsymbol{x} \tag{4.11}$$

Imposing the passivity constraint $\underline{\lambda}_{i,\mu} \geq 0$, we rewrite (4.11) as:

$$\mathbf{p}_{i,\mu}^{\mathsf{T}}\,\boldsymbol{x} \leq -\underline{\lambda}_{i,\mu} \tag{4.12}$$

with

$$\mathbf{p}_{i,\mu}^{\mathsf{T}} = -2\cdot\mathrm{Re}\left\{(\boldsymbol{v}_{i,\mu}^{\mathsf{H}}\otimes\boldsymbol{u}_{i,\mu}^{\mathsf{T}})\otimes\mathbf{a}_{i,\mu}^{\mathsf{T}}\right\} \tag{4.13}$$

Assuming to have a total number of frequency and parameter violation points equal to $\bar{\imath}$ and $\bar{\mu}$, respectively, collecting all the passivity constraints we can write (4.6) as:

$$\begin{aligned}&\min_{\boldsymbol{x}} \|\boldsymbol{\Xi}\boldsymbol{x}\|_2^2 \\ &\text{s.t. } \mathbf{P}\cdot\boldsymbol{x} \leq \boldsymbol{\lambda}\end{aligned} \tag{4.14}$$

with $\mathbf{P} = [\mathbf{p}_{0,0}^{\mathsf{T}},\ldots,\mathbf{p}_{\bar{\imath},\bar{\mu}}^{\mathsf{T}}]^{\mathsf{T}}$ and $\boldsymbol{\lambda} = [\underline{\lambda}_{0,0},\ldots,\underline{\lambda}_{\bar{\imath},\bar{\mu}}]^{\mathsf{T}}$.

4.4.2 Parametric Scattering Systems

Denoting as $\underline{\sigma}_{i,\mu}$ the largest singular value of $\mathbf{H}(\mathrm{j}\bar{\omega}_i;\theta_\mu)$, under a perturbation matrix $\Delta\mathbf{H}(\mathrm{j}\bar{\omega}_i;\vartheta)$ that acts on $\mathbf{H}(\mathrm{j}\bar{\omega}_i;\vartheta_\mu)$, $\underline{\sigma}_{i,\mu}$ will be correspondingly perturbed, at first order, toward $\widehat{\sigma}_{i,\mu}$ as shown below:

$$\widehat{\sigma}_{i,\mu} = \underline{\sigma}_{i,\mu} + \mathrm{Re}\left\{\boldsymbol{u}_{i,\mu}^{\mathsf{H}}\Delta\mathbf{H}(\mathrm{j}\bar{\omega}_{i,\mu};\theta_\mu)\boldsymbol{v}_{i,\mu}\right\} \tag{4.15}$$

where $\boldsymbol{u}_{i,\mu}$ and $\boldsymbol{v}_{i,\mu}$ are, respectively, the left and right singular vectors associated with $\underline{\sigma}_{i,\mu}$.
Recalling the definitions of $\alpha_{i,\mu;n,\ell}$ and $\mathbf{a}_{i,\mu}$, we can cast Equation (4.15) as:

$$\widehat{\sigma}_{i,\mu} = \underline{\sigma}_{i,\mu} + \mathrm{Re}\left\{(\boldsymbol{v}_{i,\mu}^{\mathsf{T}} \otimes \boldsymbol{u}_{i,\mu}^{\mathsf{H}}) \otimes \mathbf{a}_{i,\mu}^{\mathsf{T}}\right\} \boldsymbol{x} \tag{4.16}$$

Imposing the passivity constraint $\widehat{\sigma}_{i,\mu} \leq 1$, we rewrite (4.16) as:

$$\mathbf{f}_{i,\mu}^{\mathsf{T}}\ \boldsymbol{x} \leq 1 - \underline{\sigma}_{i,\mu} \tag{4.17}$$

with

$$\mathbf{f}_{i,\mu}^{\mathsf{T}} = \mathrm{Re}\left\{(\boldsymbol{v}_{i,\mu}^{\mathsf{T}} \otimes \boldsymbol{u}_{i,\mu}^{\mathsf{H}}) \otimes \mathbf{a}_{i,\mu}^{\mathsf{T}}\right\} \tag{4.18}$$

Assuming to have a total number of frequency and parameter violation points equal to $\bar{\imath}$ and $\bar{\mu}$ respectively, collecting all the passivity constraints we can write (4.6) as:

$$\begin{aligned} &\min_{\boldsymbol{x}} \|\boldsymbol{\Xi}\boldsymbol{x}\|_2^2 \\ &\text{s.t. } \mathbf{F} \cdot \boldsymbol{x} \leq \mathbf{s} \end{aligned} \tag{4.19}$$

with $\mathbf{F} = [\mathbf{f}_{0,0}^{\mathsf{T}}, \ldots, \mathbf{f}_{\bar{\imath},\bar{\mu}}^{\mathsf{T}}]^{\mathsf{T}}$ and $\mathbf{s} = [1 - \underline{\sigma}_{0,0}, \ldots, 1 - \underline{\sigma}_{\bar{\imath},\bar{\mu}}]^{\mathsf{T}}$.

Since we are using linearized local passivity constraints, the same problems discussed in Section 3.4 arise and lead to an iterative enforcement approach. Thus, even if it has been developed for non-parametric macro-models, an algorithm similar to 3.2 can be exploited to enforce passivity in the parametric case. A pseudo-code for it is given in Algorithm 4.5.

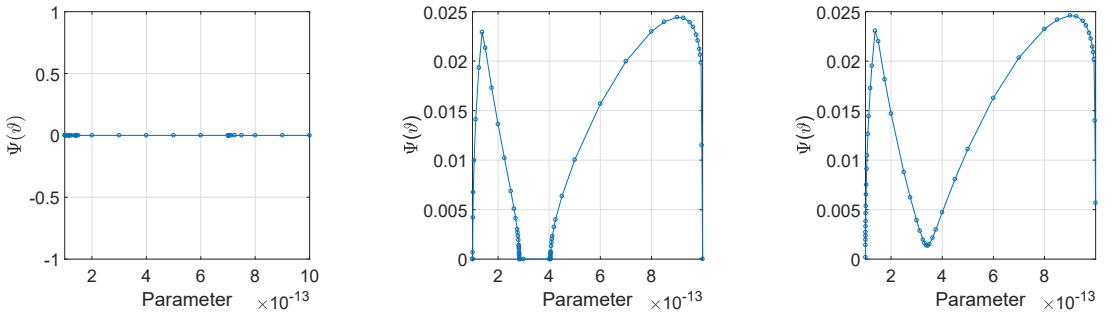
Algorithm 4.5 Passivity enforcement algorithm

Require: Model coefficients $\mathbf{R}_{n,\ell},\ r_{n,\ell}$ for $n = 1, \ldots, \bar{n},\ \ell = 1, \ldots, \bar{\ell}$
1: Get passivity violations structure $\mathcal{W}$ computed as in Algorithm 4.4
2: **while** $\mathcal{W} \neq \emptyset$ **do**
3: Build constraints as in (4.12) or (4.17)
4: Build cost function regressor $\boldsymbol{\Xi}$ as in (4.9)
5: Get residues perturbations $\Delta\mathbf{R}_{n,\ell}$ solving convex optimization problem as in (4.19) or (4.14)
6: Update model coefficients as $\mathbf{R}_{n,\ell} \leftarrow \mathbf{R}_{n,\ell} + \Delta\mathbf{R}_{n,\ell}$
7: Get passivity violations vector $\mathcal{W}$ of the perturbed model with algorithm 4.4
8: **end while**
9: **return** Passive model coefficients

4.5 Numerical Results

In this section we are going to present a set of numerical results related to passivity enforcement on parametric macro-models. In particular we will show the performances

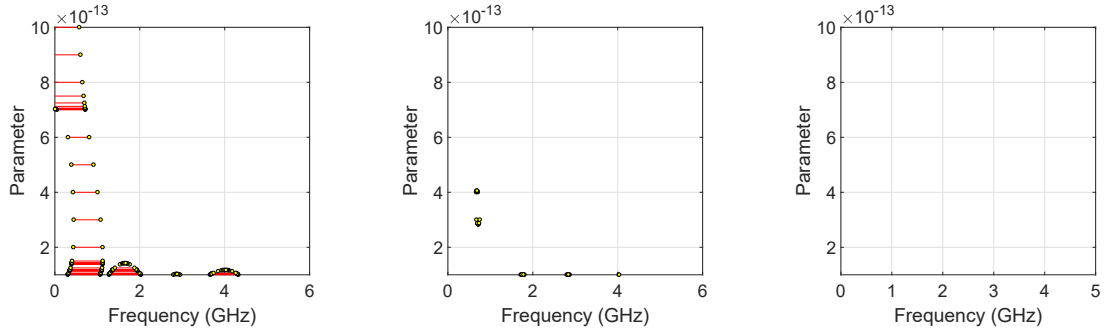
of Algorithms 4.4 and 4.5 on *Test Case 7* (see Appendix A). We start from a non passive model, whose $\Psi(\vartheta)$ function is represented in Figure 4.4a. Through the proposed passivity enforcement algorithm we present a set of plots, shown in Figures from 4.4a to 4.4c, representing $\Psi(\vartheta)$ associated with models at successive enforcement iterations, until passivity is reached. We see that, as the number of iterations grow, $\Psi(\vartheta)$ is "lifted up", until it is uniformly strictly positive, which means that the model is passive for all the feasible parameter points. In Figures from 4.5a to 4.5c we look at the passivity



(a) $\Psi(\vartheta)$ at first iteration (b) $\Psi(\vartheta)$ at second iteration (c) $\Psi(\vartheta)$ at final iteration

Figure 4.4: $\Psi(\vartheta)$ at successive enforcement iterations (*Test Case 7*)

enforcement process from another standpoint. In fact we are looking at the (ω, ϑ) plane: the yellow dots represent the passivity violations (i.e., the frequencies corresponding to purely imaginary Hamiltonian eigenvalues) found with the Hamiltonian driven scheme ad the red lines are the non passive frequency sub-bands. We see that, as before, as the number of iterations grow, violation areas disappear, until we have a passive model where no violations are detected.



(a) Violations in (ω, ϑ) plane (first iteration) (b) Violations in (ω, ϑ) plane (second iteration) (c) Violations in (ω, ϑ) plane (final iteration)

Figure 4.5: Violation in frequency-parameter plane at successive enforcement iterations (*Test Case 7*)

About the accuracy with respect to initial data, Figure 4.6 shows the frequency response of the passive model with respect to tabulated data for a set of parameter values. Wee see that, throughout the bandwidth and for all the represented parameter values, the model

is still very accurate. Quantitatively the maximum relative error is $4.5 \cdot 10^{-3}$, that, even if higher with respect to the non-passive model one, is negligible for our purposes.

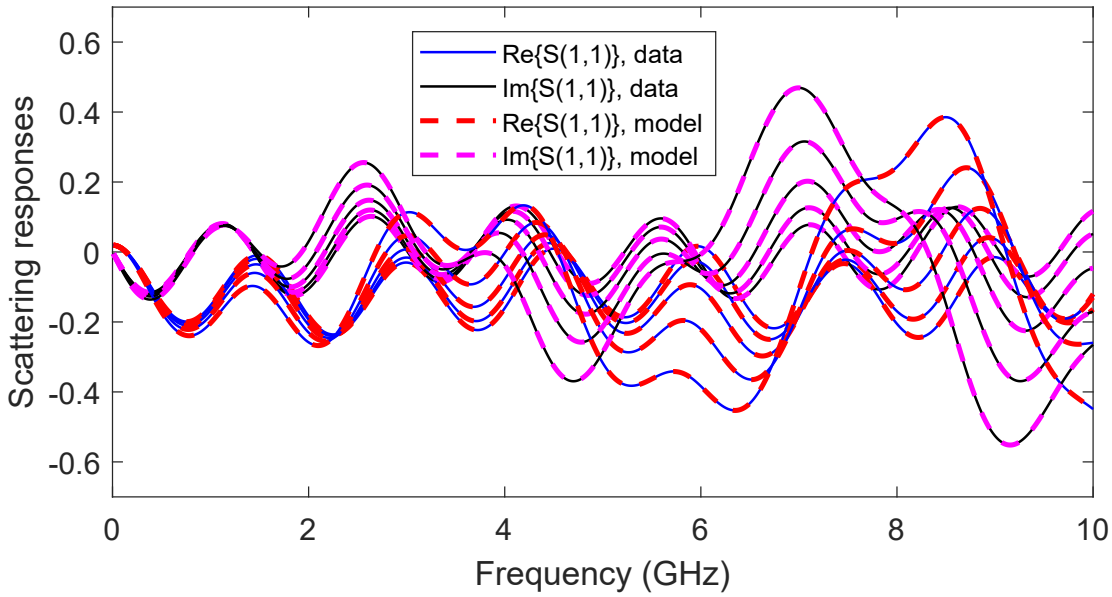


Figure 4.6: Data and passive model response comparison (*Test Case 7*)

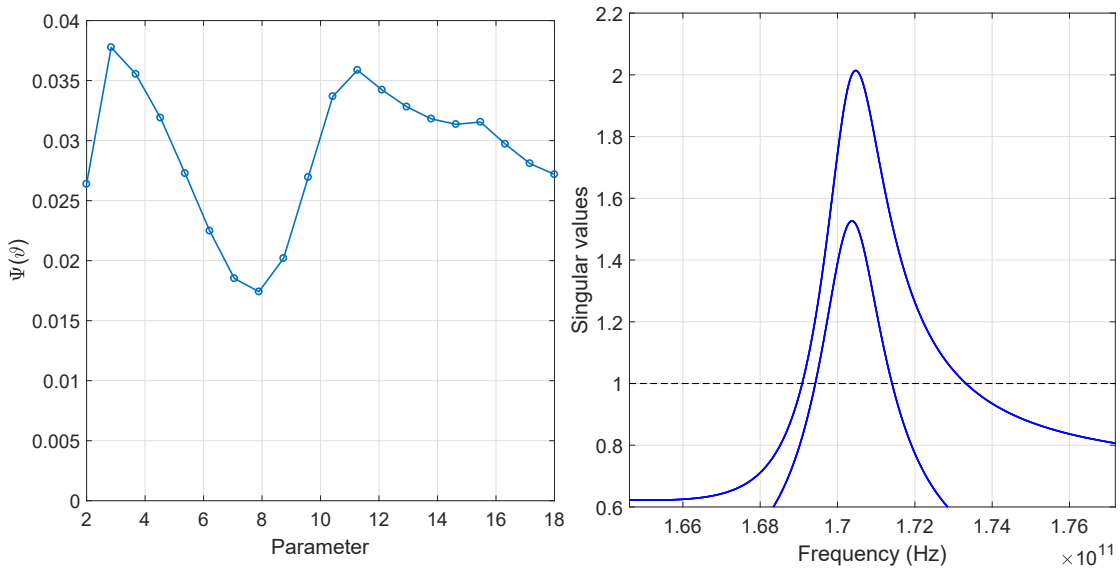
Chapter 5

A Linear-Prediction Based Passivity Verification Scheme

In this chapter we introduce a novel adaptive passivity verification method. The necessity for a new passivity assessment algorithm comes from the need of both a higher reliability and a scalable approach to high order multi-variate macro-models passivity enforcement. Here we introduce the mono-parametric case, while in Chapter 6 we will extend it to the multi-variate case.
Even if the passivity verification method presented in Chapter 4 (from now on defined as Ψ-*based*) turns out to be effective in most of the cases, there are some critical circumstances in which it may fail in detecting some violation areas. In fact, Figure 5.1a shows the function $\Psi(\vartheta)$, defined in (4.2), for a model (whose structure is detailed in Appendix A, under *Test Case 2–b*) made passive with the Ψ-based approach. Thus we expect that, for any parameter value in the parameter space Θ, there are not singular values greater than 1. However, if we evaluate our passive model for $\vartheta = 7.84 \in \Theta$, and we perform a brute-force singular values sampling (the model is in scattering representation) we see that, as Figure 5.1b shows, for a frequency $f \approx 27$ GHz, there is a violation. This means that the Ψ-based scheme defines as passive a non-passive model.
This problem is related to the "blindness" of the Ψ-based algorithm to fast Hamiltonian eigenvalues variations inside a parameter space sub-interval. To overcome this problem we propose an algorithm based on linear perturbations of Hamiltonian eigenvalues that is able to predict, at first order, where after a small parameter variation these eigenvalues fall, without explicitly computing them.
In the following, after a brief introduction to matrix perturbation theory, we extensively present, with the help of several numerical results, this novel passivity assessment algorithm. Finally, we will discuss in detail about asymptotic passivity for parameterized macro-models.

5.1 Eigenvalue Perturbations

In this chapter the concepts of eigenvalues perturbations and polynomial derivatives are extensively used. This section covers some basic aspects of these topics. The problem



(a) $\Psi(\vartheta)$ adaptively sampled by the Ψ-based algorithm (*Test Case 2–b*)

(b) Violation on singular values/frequency plane (*Test Case 2–b*)

Figure 5.1

we face here is to find the eigenvalues of a matrix, or a matrix pencil, when the matrix coefficients are perturbed by an infinitesimal quantity. Matrix perturbation theory shows that, for a given parameter dependent matrix $\mathbf{M}(\vartheta)$ or for a $(\mathbf{M}(\vartheta), \mathbf{K})$ pencil, it is possible to retrieve information about eigenvalues positions of a new perturbed matrix, obtained as $\mathbf{M}(\vartheta + d\vartheta) \approx \mathbf{M} + d\mathbf{M}$, with $d\vartheta$ an infinitesimal perturbation. Considering an initial parameter value $\vartheta = \vartheta_0$, the matrix $d\mathbf{M}(\vartheta_0)$ reads:

$$d\mathbf{M}(\vartheta_0) = \frac{\partial \mathbf{M}(\vartheta)}{\partial \vartheta}|_{\vartheta=\vartheta_0} \cdot d\vartheta$$

The eigenvalues displacements, indicated as $d\lambda_{\vartheta_0}$, induced by an infinitesimal perturbation $d\mathbf{M}(\vartheta_0)$ on a matrix $\mathbf{M}(\vartheta_0)$ can be computed analytically as:

$$d\lambda_{\vartheta_0} = \frac{\partial \lambda(\vartheta)}{\partial \vartheta}|_{\vartheta=\vartheta_0} \cdot d\vartheta \tag{5.1}$$

where

$$d\lambda_{\vartheta_0} = \frac{\boldsymbol{v}^{\mathsf{H}}(\vartheta_0) \cdot d\mathbf{M}(\vartheta_0) \cdot \boldsymbol{u}(\vartheta_0)}{\boldsymbol{v}^{\mathsf{H}}(\vartheta_0) \cdot \boldsymbol{u}(\vartheta_0)} \tag{5.2}$$

while for a pencil $(\mathbf{M}(\vartheta_0), \mathbf{K})$ it reads:

$$d\lambda_{\vartheta_0} = \frac{\boldsymbol{v}^{\mathsf{H}}(\vartheta_0) \cdot d\mathbf{M}(\vartheta_0) \cdot \boldsymbol{u}(\vartheta_0)}{\boldsymbol{v}^{\mathsf{H}}(\vartheta_0) \cdot \mathbf{K} \cdot \boldsymbol{u}(\vartheta_0)} \tag{5.3}$$

where $\boldsymbol{v}(\vartheta_0)$ and $\boldsymbol{u}(\vartheta_0)$ are respectively the right and left matrix/pencil eigenvectors

computed at parameter value ϑ_0. For further details on matrix pencils perturbation see [43].
Eigenvalues perturbation techniques will be used extensively for Hamiltonian matrices associated with the system realization, that have been derived in Section 1.6.2. Thus, in the following, we will discuss, step by step, how to compute analytically Hamiltonian perturbed eigenvalues. To this end, we must derive an expression for $d\mathbf{M}$, for Hamiltonian matrices associated with state-space and descriptor realizations.

5.1.1 State-space Realizations

The state-space realization for a system transfer matrix $\mathbf{H}(s;\vartheta)$ is detailed in Section 2.1.2. In order to derive an expression for $d\mathbf{M}$, the state-space realization matrices must be differentiated with respect to the parameter (from now on, the dependency on the parameter has been omitted for readability):

$$\frac{\partial \mathbf{A}}{\partial \vartheta} = \mathbf{B_0}\frac{\partial \mathbf{D}_2^{-1}}{\partial \vartheta}\mathbf{C_2} - \mathbf{B}_0\mathbf{D}_2^{-1}\frac{\partial \mathbf{C}_2}{\partial \vartheta} \tag{5.4}$$

$$\frac{\partial \mathbf{B}}{\partial \vartheta} = \mathbf{B}_0\frac{\partial \mathbf{D}_2^{-1}}{\partial \vartheta}$$

$$\frac{\partial \mathbf{C}}{\partial \vartheta} = \frac{\partial \mathbf{C}_1}{\partial \vartheta} - \frac{\partial \mathbf{D}_1}{\partial \vartheta}\mathbf{D}_2^{-1}\mathbf{C_2} - \mathbf{D}_1\frac{\partial \mathbf{D}_2^{-1}}{\partial \vartheta}\mathbf{C}_2 - \mathbf{D}_1\mathbf{D}_2^{-1}\frac{\partial \mathbf{C}_2}{\partial \vartheta}$$

$$\frac{\partial \mathbf{D}}{\partial \vartheta} = \frac{\partial \mathbf{D}_1}{\partial \vartheta}\mathbf{D}_2^{-1} + \mathbf{D}_1\frac{\partial \mathbf{D}_2^{-1}}{\partial \vartheta}$$

where

$$\frac{\partial \mathbf{D}_1^{-1}}{\partial \vartheta} = \frac{\partial \mathbf{R}_0^{-1}(\vartheta)}{\partial \vartheta}$$

$$\frac{\partial \mathbf{D}_2^{-1}}{\partial \vartheta} = \mathbb{I}_P\frac{\partial}{\partial \vartheta}\left(\frac{1}{r_0(\vartheta)}\right)$$

For convenience, we recall from Section 1.6 the Hamiltonian matrices associated with immittance ($\boldsymbol{\mathcal{N}}_0$) and scattering ($\boldsymbol{\mathcal{M}}_1$) systems in state-space representation:

$$\boldsymbol{\mathcal{N}}_0 = \begin{pmatrix} \mathbf{A}-\mathbf{B}(\mathbf{D}+\mathbf{D}^\mathsf{T})^{-1}\mathbf{C} & -\mathbf{B}(\mathbf{D}+\mathbf{D}^\mathsf{T})^{-1}\mathbf{B}^\mathsf{T} \\ \mathbf{C}^\mathsf{T}(\mathbf{D}+\mathbf{D}^\mathsf{T})^{-1}\mathbf{C} & -\mathbf{A}^\mathsf{T}+\mathbf{C}^\mathsf{T}(\mathbf{D}+\mathbf{D}^\mathsf{T})^{-1}\mathbf{B}^\mathsf{T} \end{pmatrix}$$

$$\boldsymbol{\mathcal{M}}_1 = \begin{pmatrix} \mathbf{A}+\mathbf{B}(\mathbb{I}-\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{D}^\mathsf{T}\mathbf{C} & \mathbf{B}(\mathbb{I}-\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{B}^\mathsf{T} \\ -\mathbf{C}^\mathsf{T}(\mathbb{I}-\mathbf{D}\mathbf{D}^\mathsf{T})^{-1}\mathbf{C} & -\mathbf{A}^\mathsf{T}-\mathbf{C}^\mathsf{T}\mathbf{D}(\mathbb{I}-\mathbf{D}^\mathsf{T}\mathbf{D})^{-1}\mathbf{B}^\mathsf{T} \end{pmatrix}$$

By differentiating each term with respect to the parameter we obtain, for immittance systems:

$$\frac{\partial \boldsymbol{\mathcal{N}}_0}{\partial \vartheta} = \begin{pmatrix} \frac{\partial \mathbf{A}}{\partial \vartheta} - \frac{\partial}{\partial \vartheta}(\mathbf{B}\mathbf{Q}_0^{-1}\mathbf{C}) & -\frac{\partial \mathbf{B}}{\partial \vartheta}\mathbf{Q}_0^{-1}\mathbf{B}^{\mathsf{T}} - \mathbf{B}\frac{\partial \mathbf{Q}_0^{-1}}{\partial \vartheta}\mathbf{B}^{\mathsf{T}} - \mathbf{B}\mathbf{Q}_0^{-1}\frac{\partial \mathbf{B}^{\mathsf{T}}}{\partial \vartheta} \\ \frac{\partial \mathbf{C}^{\mathsf{T}}}{\partial \vartheta}\mathbf{Q}_0^{-1}\mathbf{C} + \mathbf{C}^{\mathsf{T}}\frac{\partial \mathbf{Q}_0^{-1}}{\partial \vartheta}\mathbf{C} + \mathbf{C}^{\mathsf{T}}\mathbf{Q}_0^{-1}\frac{\partial \mathbf{C}}{\partial \vartheta} & -\frac{\partial \mathbf{A}^{\mathsf{T}}}{\partial \vartheta} + \frac{\partial}{\partial \vartheta}(\mathbf{C}^{\mathsf{T}}\mathbf{Q}_0^{-1}\mathbf{B}^{\mathsf{T}}) \end{pmatrix}, \tag{5.5}$$

$\mathbf{Q}_0 = \mathbf{D} + \mathbf{D}^{\mathsf{T}}$

with

$$\frac{\partial}{\partial \vartheta}(\mathbf{B}\mathbf{Q}_0^{-1}\mathbf{C}) = \frac{\partial \mathbf{B}}{\partial \vartheta}\mathbf{Q}_0^{-1}\mathbf{C} + \mathbf{B}\frac{\partial \mathbf{Q}_0^{-1}}{\partial \vartheta}\mathbf{C} + \mathbf{B}\mathbf{Q}_0^{-1}\frac{\partial \mathbf{C}}{\partial \vartheta}$$
$$\frac{\partial}{\partial \vartheta}(\mathbf{C}^{\mathsf{T}}\mathbf{Q}_0^{-1}\mathbf{B}^{\mathsf{T}}) = \frac{\partial \mathbf{C}^{\mathsf{T}}}{\partial \vartheta}\mathbf{Q}_0^{-1}\mathbf{B}^{\mathsf{T}} + \mathbf{C}^{\mathsf{T}}\frac{\partial \mathbf{Q}_0^{-1}}{\partial \vartheta}\mathbf{B}^{\mathsf{T}} + \mathbf{C}^{\mathsf{T}}\mathbf{Q}_0^{-1}\frac{\partial \mathbf{B}^{\mathsf{T}}}{\partial \vartheta}$$

while for scattering systems we have:

$$\frac{\partial \boldsymbol{\mathcal{M}}_1}{\partial \vartheta} = \begin{pmatrix} \frac{\partial \mathbf{A}}{\partial \vartheta} - \frac{\partial}{\partial \vartheta}(\mathbf{B}\mathbf{Q}_1^{-1}\mathbf{D}^{\mathsf{T}}\mathbf{C}) & \frac{\partial \mathbf{B}}{\partial \vartheta}\mathbf{Q}_1^{-1}\mathbf{B}^{\mathsf{T}} + \mathbf{B}\frac{\partial \mathbf{Q}_1^{-1}}{\partial \vartheta}\mathbf{B}^{\mathsf{T}} + \mathbf{B}\mathbf{Q}_1^{-1}\frac{\partial \mathbf{B}^{\mathsf{T}}}{\partial \vartheta} \\ -\frac{\partial \mathbf{C}^{\mathsf{T}}}{\partial \vartheta}\widetilde{\mathbf{Q}}_1^{-1}\mathbf{C} - \mathbf{C}^{\mathsf{T}}\frac{\partial \widetilde{\mathbf{Q}}_1^{-1}}{\partial \vartheta}\mathbf{C} - \mathbf{C}^{\mathsf{T}}\widetilde{\mathbf{Q}}_1^{-1}\frac{\partial \mathbf{C}}{\partial \vartheta} & -\frac{\partial \mathbf{A}^{\mathsf{T}}}{\partial \vartheta} - \frac{\partial}{\partial \vartheta}(\mathbf{C}^{\mathsf{T}}\mathbf{D}\mathbf{Q}_1^{-1}\mathbf{B}^{\mathsf{T}}) \end{pmatrix}, \tag{5.6}$$

$\mathbf{Q}_1 = \mathbb{I} - \mathbf{D}^{\mathsf{T}}\mathbf{D}$
$\widetilde{\mathbf{Q}}_1 = \mathbb{I} - \mathbf{D}\mathbf{D}^{\mathsf{T}}$

where

$$\frac{\partial}{\partial \vartheta}(\mathbf{B}\mathbf{Q_1}^{-1}\mathbf{D}^{\mathsf{T}}\mathbf{C}) = \frac{\partial \mathbf{B}}{\partial \vartheta}\mathbf{Q}_1^{-1}\mathbf{D}^{\mathsf{T}}\mathbf{C} + \mathbf{B}\frac{\partial \mathbf{Q}_1^{-1}}{\partial \vartheta}\mathbf{D}^{\mathsf{T}}\mathbf{C} + \mathbf{B}\mathbf{Q}_1^{-1}\frac{\partial \mathbf{D}^{\mathsf{T}}}{\partial \vartheta}\mathbf{C} + \mathbf{B}\mathbf{Q}_1^{-1}\mathbf{D}^{\mathsf{T}}\frac{\partial \mathbf{C}}{\partial \vartheta}$$

$$\frac{\partial}{\partial \vartheta}(\mathbf{C}^{\mathsf{T}}\mathbf{D}\mathbf{Q}_1^{-1}\mathbf{B}^{\mathsf{T}}) = \frac{\partial \mathbf{C}^{\mathsf{T}}}{\partial \vartheta}\mathbf{D}\mathbf{Q}_1^{-1}\mathbf{B}^{\mathsf{T}} + \mathbf{C}^{\mathsf{T}}\frac{\partial \mathbf{D}}{\partial \vartheta}\mathbf{Q}_1^{-1}\mathbf{B}^{\mathsf{T}} + \mathbf{C}^{\mathsf{T}}\mathbf{D}\frac{\partial \mathbf{Q}_1^{\mathsf{T}}}{\partial \vartheta}\mathbf{B}^{\mathsf{T}} + \mathbf{C}^{\mathsf{T}}\mathbf{D}\mathbf{Q}_1^{-1}\frac{\partial \mathbf{B}^{\mathsf{T}}}{\partial \vartheta}$$

5.1.2 Descriptor Systems

The descriptor realization for a system described by a transfer matrix $\mathbf{H}(s;\vartheta)$, is detailed in Section 2.1.2. As for the state-space case, we must compute the first order derivatives

of descriptor matrices, as:

$$\frac{\partial \mathbf{E}}{\partial \vartheta} = \mathbf{0}_{N+P} \tag{5.7}$$

$$\frac{\partial \mathbf{A}}{\partial \vartheta} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \frac{\partial \mathbf{C}_2}{\partial \vartheta} & \frac{\partial \mathbf{D}_2}{\partial \vartheta} \end{pmatrix}$$

$$\frac{\partial \mathbf{B}}{\partial \vartheta} = \begin{pmatrix} \mathbf{0}_{N,P} \\ \mathbf{0}_P \end{pmatrix}$$

$$\frac{\partial \mathbf{C}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial \mathbf{C}_1}{\partial \vartheta}, & \frac{\partial \mathbf{D}_1}{\partial \vartheta} \end{pmatrix}$$

For convenience we recall from Section 1.6 the structure of Hamiltonian matrices for immittance ($\boldsymbol{\mathcal{N}}_0^{ext}$) and scattering ($\boldsymbol{\mathcal{M}}_1^{ext}$) systems in descriptor realization:

$$\boldsymbol{\mathcal{N}}_0^{ext} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^\mathsf{T} & -\mathbf{C}^\mathsf{T} \\ \mathbf{C} & \mathbf{B}^\mathsf{T} & \mathbf{D}+\mathbf{D}^\mathsf{T} \end{pmatrix}, \qquad \boldsymbol{\mathcal{K}} = \begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^\mathsf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

$$\boldsymbol{\mathcal{M}}_1^{ext} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}^\mathsf{T} & \mathbf{0} & -\mathbf{C}^\mathsf{T} \\ \mathbf{0} & \mathbf{B}^\mathsf{T} & -\mathbb{I} & \mathbf{D}^\mathsf{T} \\ \mathbf{C} & \mathbf{0} & \mathbf{D} & \mathbb{I} \end{pmatrix}, \qquad \boldsymbol{\mathcal{K}} = \begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^\mathsf{T} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

By differentiating these matrices we get, for immittance systems:

$$\frac{\partial \boldsymbol{\mathcal{N}}_0^{ext}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial \mathbf{A}}{\partial \vartheta} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\frac{\partial \mathbf{A}^\mathsf{T}}{\partial \vartheta} & -\frac{\partial \mathbf{C}^\mathsf{T}}{\partial \vartheta} \\ \frac{\partial \mathbf{C}}{\partial \vartheta} & \mathbf{0} & \frac{\partial \mathbf{D}}{\partial \vartheta} + \frac{\partial \mathbf{D}^\mathsf{T}}{\partial \vartheta} \end{pmatrix} \tag{5.8}$$

while for scattering systems we have:

$$\frac{\partial \boldsymbol{\mathcal{M}}_1^{ext}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial \mathbf{A}}{\partial \vartheta} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\frac{\partial \mathbf{A}^\mathsf{T}}{\partial \vartheta} & \mathbf{0} & -\frac{\partial \mathbf{C}^\mathsf{T}}{\partial \vartheta} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{D}^\mathsf{T}}{\partial \vartheta} \\ \frac{\partial \mathbf{C}}{\partial \vartheta} & \mathbf{0} & \frac{\partial \mathbf{D}}{\partial \vartheta} & \mathbf{0} \end{pmatrix} \tag{5.9}$$

By looking at realization matrices definitions we see that, to obtain their derivatives, we need to differentiate the numerator and denominator residues contained in $\mathbf{C}_1(\vartheta)$, $\mathbf{C}_2(\vartheta)$, $\mathbf{D}_1(\vartheta)$ and $\mathbf{D}_2(\vartheta)$ (as shown in Section 2.1.2) and assemble them properly.
Recalling that the numerator and denominator residues associated n-th order partial fraction are:

$$\mathbf{R}_n(\vartheta) = \sum_{\ell=0}^{\bar{\ell}} \mathbf{R}_{n,\ell}\xi_\ell(\vartheta) \qquad\qquad r_n(\vartheta) = \sum_{\ell=0}^{\bar{\ell}} r_{n,\ell}\xi_\ell(\vartheta) \tag{5.10}$$

it holds that the differentiated residues have the form:

$$\frac{\partial \mathbf{R}_n(\vartheta)}{\partial \vartheta} = \sum_{\ell=0}^{\bar{\ell}} \mathbf{R}_{n,\ell}\frac{\partial \xi_\ell(\vartheta)}{\partial \vartheta} \qquad\qquad \frac{\partial r_n(\vartheta)}{\partial \vartheta} = \sum_{\ell=0}^{\bar{\ell}} r_{n,\ell}\frac{\partial \xi_\ell(\vartheta)}{\partial \vartheta} \tag{5.11}$$

Equation (5.11) shows that, to get differentiated residues, it suffices to compute the basis functions derivatives and then multiply them with the corresponding residues $\mathbf{R}_{n,\ell}$, $r_{n,\ell}$.

Thus, in order to get the matrix $\frac{\partial \mathbf{M}}{\partial \vartheta}|_{\vartheta=\vartheta_0}$ we follow this procedure:

1. All the basis function $\xi_\ell(\vartheta)$ are differentiated with respect to the parameter and evaluated for $\vartheta = \vartheta_0$;
2. The differentiated residues at ϑ_0 are computed through (5.11);
3. With the differentiated residues we construct the differentiated realization matrices through (5.4) and (5.7);
4. Finally the matrix $\frac{\partial \mathbf{M}}{\partial \vartheta}|_{\vartheta=\vartheta_0}$ is re-constructed through (5.5), (5.6), (5.8) and (5.9).

In the next section, we address the problem of computing basis function derivatives in the case where these functions are polynomials.

5.1.3 Basis Functions Derivatives

As detailed in the previous section, to get differentiated Hamiltonian matrices we proceed by computing numerator and denominator residues derivatives. The main idea that underlies this computation is based on the particular structure of parameter basis functions, that most of the times are polynomials in the variable ϑ. In these cases, what relates basis functions and their derivatives are just linear transformations.
Thus, considering $\xi_\ell(\vartheta)$ as a generic ℓ-th order polynomial parameter basis function, we will set:

$$\mathbf{b}_{\bar{\ell}}(\vartheta) = \begin{pmatrix} \xi_1(\vartheta) \\ \xi_2(\vartheta) \\ \vdots \\ \xi_{\bar{\ell}}(\vartheta) \end{pmatrix}$$

It is then possible to write the vector of differentiated polynomials as:

$$\frac{\partial \mathbf{b}_{\bar{\ell}}(\vartheta)}{\partial \vartheta} = \mathbf{T}_{\bar{\ell}} \cdot \mathbf{b}_{\bar{\ell}}(\vartheta)$$

where $\mathbf{T}_{\bar{\ell}} \in \mathbb{R}^{\bar{\ell}\times\bar{\ell}}$ is a suitable $\bar{\ell}$-th order transformation matrix (how this matrix is computed is detailed is Section 5.1.4).
Assuming that we are working with a mono-parametric model (multivariate cases can be reduced to mono-parametric by evaluating the basis functions for all parameters except the one for which the derivative is computed), we denote as $\mathbf{r} \in \mathbb{C}^{\bar{n}\bar{\ell}\times 1}$ the vector collecting the denominator residues. It is possible to re-organize its coefficients as a matrix $\overline{\mathbf{R}} \in \mathbb{R}^{\bar{n}\times\bar{\ell}}$ as shown below:

$$\mathbf{r} = \begin{bmatrix} r_{00} \\ \vdots \\ r_{\bar{n}0} \\ \vdots \\ r_{0\bar{\ell}} \\ \vdots \\ r_{\bar{n}\bar{\ell}} \end{bmatrix} \qquad \overline{\mathbf{R}} = \begin{bmatrix} r_{00} & \cdots & r_{0\bar{\ell}} \\ \vdots & & \vdots \\ r_{\bar{n}0} & \cdots & r_{\bar{n}\bar{\ell}} \end{bmatrix}$$

This re-shaping operation is formally written as:

$$\overline{\mathbf{R}} = \mathrm{mat}(\mathbf{r})$$

where the operator "mat$(\cdot)$" reshapes a $[\bar{n}\bar{l}\times 1]$ vector as a $[\bar{n}\times\bar{l}]$ matrix.

The scalar product between the n-th row of $\overline{\mathbf{R}}$ and the column vector $\mathbf{b}_{\bar{\ell}}$ is exactly the same as (5.10). Thus, Equation (5.12) returns a vector $\mathbf{r}(\vartheta) \in \mathbb{R}^{\bar{n}\times 1}$ that contains the residues associated to denominator poles, evaluated for a given parameter value ϑ.

$$\mathbf{r}(\vartheta) = \overline{\mathbf{R}}\ \mathbf{b}_{\bar{\ell}}(\vartheta) \tag{5.12}$$

Supposing that the transformation matrix $\mathbf{T}_{\bar{\ell}}$ is known, the differentiated model denominator residues $\frac{\partial \mathbf{r}(\vartheta)}{\partial \vartheta} \in \mathbb{R}^{\bar{n}}$ can be computed, recalling (5.11) as:

$$\frac{\partial \mathbf{r}(\vartheta)}{\partial \vartheta} = \overline{\mathbf{R}} \cdot \frac{\partial \mathbf{b}_{\bar{\ell}}(\vartheta)}{\partial \vartheta} = \overline{\mathbf{R}} \cdot \mathbf{T}_{\bar{\ell}} \cdot \mathbf{b}_{\bar{\ell}}(\vartheta)$$

The same procedure is applied to numerator residues, by considering one input/output port combination at a time.

In the next section we will derive transformation matrices $\mathbf{T}_{\ell}$ for monomial and Chebyshev parameter basis functions.

5.1.4 Linear Transformation Matrices

Monomials Parameter Basis

The monomial basis is defined as a set of monomials in ϑ, up to the $\bar{\ell}$-th order.
The basis vector $\mathbf{b}_{\bar{\ell}}$ thus reads:

$$\mathbf{b}_{\bar{\ell}}(\vartheta) = \begin{pmatrix} \vartheta^0 \\ \vdots \\ \vartheta^{\bar{\ell}} \end{pmatrix}$$

and the linear transformation is associated with a matrix $\mathbf{T}_{\bar{\ell}}$ whose structure is:

$$\mathbf{T}_{\bar{\ell}} = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 3 & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \bar{\ell} & 0 \end{pmatrix}$$

The derivation comes from the simple differentiation rule for monomials.

Chebyshev Parameter Basis

The derivative of a ℓ-th order Chebyshev polynomial (first kind), here indicated as $T_\ell(\vartheta)$, can be computed through this recursive formula

$$\frac{\partial T_\ell(\vartheta)}{\partial \vartheta} = \ell \left(2T_{\ell-1}(\vartheta) + \frac{1}{\ell - 2} \frac{\partial T_{\ell-2}(\vartheta)}{\partial \vartheta} \right) \tag{5.13}$$

This relation shows that Chebyshev polynomial derivatives are linear combination of Chebyshev polynomials themselves. Thus, we have a rule to recursively compute these derivatives that can be easily implemented as a recursive algorithm.
First of all, it must be noticed that this formula is applicable just for $\ell > 2$, so the transformation matrix $\mathbf{T}_\ell \in \mathbb{Z}^{(\ell+1)\times(\ell+1)}$ must be initialized by hand, at least up to second order.
For Chebyshev polynomials these first order derivatives are:

$$\begin{aligned} T_0(\vartheta) &= 1 & \frac{\partial T_0(\vartheta)}{\partial \vartheta} &= 0 \\ T_1(\vartheta) &= \vartheta & \frac{\partial T_1(\vartheta)}{\partial \vartheta} &= 1 \\ T_2(\vartheta) &= 2\vartheta^2 - 1 & \frac{\partial T_2(\vartheta)}{\partial \vartheta} &= 4\vartheta \end{aligned} \tag{5.14}$$

Derivatives in (5.14) can be written as linear combinations of $T_0(\vartheta)$, $T_1(\vartheta)$ and $T_2(\vartheta)$ as:

$$\begin{cases} \frac{\partial T_0(\vartheta)}{\partial \vartheta} = 0 \cdot T_0(\vartheta) + 0 \cdot T_1(\vartheta) + 0 \cdot T_2(\vartheta) \\ \frac{\partial T_1(\vartheta)}{\partial \vartheta} = 1 \cdot T_0(\vartheta) + 0 \cdot T_1(\vartheta) + 0 \cdot T_2(\vartheta) \\ \frac{\partial T_2(\vartheta)}{\partial \vartheta} = 0 \cdot T_0(\vartheta) + 4 \cdot T_1(\vartheta) + 0 \cdot T_2(\vartheta) \end{cases} \tag{5.15}$$

Equations in (5.15) are now cast in a matrix form:

$$\begin{pmatrix} \frac{\partial T_0(\vartheta)}{\partial \vartheta} \\ \frac{\partial T_1(\vartheta)}{\partial \vartheta} \\ \frac{\partial T_2(\vartheta)}{\partial \vartheta} \end{pmatrix} = \mathbf{T}_2 \begin{pmatrix} T_0(\vartheta) \\ T_1(\vartheta) \\ T_2(\vartheta) \end{pmatrix}$$

where $\mathbf{T}_2$ is the initial transformation matrix that reads:

$$\mathbf{T}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 4 & 0 \end{bmatrix}$$

This initial transformation matrix, valid just up to second order Chebyshev polynomials, is the starting point to build the general one. To obtain the next rows of the transformation matrix we will recursively use Equation 5.13.

With the following example we shown how to get the fourth row of $\mathbf{T_3}$ starting from $\mathbf{T}_2$. High order rows are computed using the same method.

Applying (5.13) with $\ell = 3$ leads to:

$$\frac{\partial T_3(\vartheta)}{\partial \vartheta} = 3\left(2T_2(\vartheta) + \frac{\partial T_1(\vartheta)}{\partial \vartheta}\right)$$

where the term $\frac{\partial T_1(\vartheta)}{\partial \vartheta}$ is itself a linear combination of chebyshev polynomials and can be written starting from $\mathbf{T}_2$ by considering that its 2^{nd} row corresponds to $\frac{\partial T_1(\vartheta)}{\partial \vartheta} = T_0(\vartheta)$. Thus, the previous relation can be written as:

$$\frac{\partial T_3(\vartheta)}{\partial \vartheta} = 3(2T_2(\vartheta) + T_0(\vartheta)) = 3T_0(\vartheta) + 6T_2(\vartheta)$$

Then, the fourth row of $\mathbf{T_3}$ is [3 0 6 0] and the final matrix is:

$$\mathbf{T_3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 3 & 0 & 6 & 0 \end{bmatrix}$$

Thus, in general, the ℓ-th row of the transformation matrix, denoted as $\mathbf{t}_\ell$ can be computed as:

$$\mathbf{t}_\ell = \mathbf{z}_\ell + \mathbf{a}_\ell$$

with $\mathbf{z}_\ell = [0, \ldots, \ 0, \ 2\ell, \ 0] \in \mathbb{R}^\ell$ and $\mathbf{a} = [\frac{\ell}{\ell-2}\mathbf{t}_\ell, \ 0]$.

In this section, we derived a simple method to compute the eigenvalues of a parameter dependent matrix (structurally Hamiltonian) under an infinitesimal parameter variation. In the next section we will discuss about eigenvalues perturbations induced by finite parameter variations, that are the main tool for the proposed passivity verification scheme.

5.1.5 Hamiltonian Eigenvalue Trajectories

Up to now, just infinitesimal parameter perturbations have been considered. However, if our purpose is to predict where Hamiltonian eigenvalues fall after a finite parameter variation, as usually is, some remarks are required. In fact, the eigenvalues trajectories under finite parameter variations are non-linear and hardly predictable. Thus, being able to compute the first derivative of these trajectories, as detailed in the previous section, we can make just a linear prediction on where perturbed eigenvalues will fall under a finite parameter variation.
These linearly perturbed eigenvalues can be seen as a first order Taylor expansion of an unknown non-linear function $\lambda = \lambda(\vartheta)$. Thus, denoting the finite parameter variation as $\delta\vartheta = \vartheta - \vartheta_0$, it is possible to get these perturbed eigenvalues as:

$$\widehat{\lambda}(\vartheta) = \lambda(\vartheta_0) + \frac{\partial\lambda(\vartheta)}{\partial\vartheta}|_{\vartheta=\vartheta_0} \cdot \delta\vartheta \tag{5.16}$$

as detailed in Equation (5.1).

To better explain these concepts, some numerical examples are reported. In particular, Figure 5.2 compares the exact Hamiltonian eigen-spectrum, shown in blue, computed for increasing parameter values, with the approximated one, computed with (5.16), represented in red. As one can expect, the linearly perturbed eigenvalues are straight lines centered at $\lambda(\mathbf{M}(\vartheta_0))$, while the exact ones have non-linear trajectories. In general, we have the best approximation for small parameter perturbations. This consideration is fundamental for the adaptive verification algorithm that will be presented in the next section.

Later on in this work, eigenvalue perturbations will be extensively used thus, in order to improve the readability, we introduce a compact notation to identify matrix/pencil eigenvalues perturbation.
We will denote the set of linearly perturbed eigenvalues of a matrix $\mathbf{M}(\vartheta_0)$ or pencil $(\mathbf{M}(\vartheta_0), \mathbf{K})$ in $\vartheta = \vartheta_0 + \delta\vartheta$ as:

$$\widehat{\Lambda}_{\vartheta_0}(\vartheta) = \mathcal{P}^{\delta\vartheta}_{M,K}(\vartheta_0) \tag{5.17}$$

where $\mathcal{P}^{\delta\vartheta}_{M,K}(\vartheta_0)$ is the first order perturbation operator applied on matrix $\mathbf{M}(\vartheta_0)$ or $(\mathbf{M}(\vartheta_0), \mathbf{K})$ pencil, centered at $\vartheta = \vartheta_0$, for a finite parameter variation equal to $\delta\vartheta$.

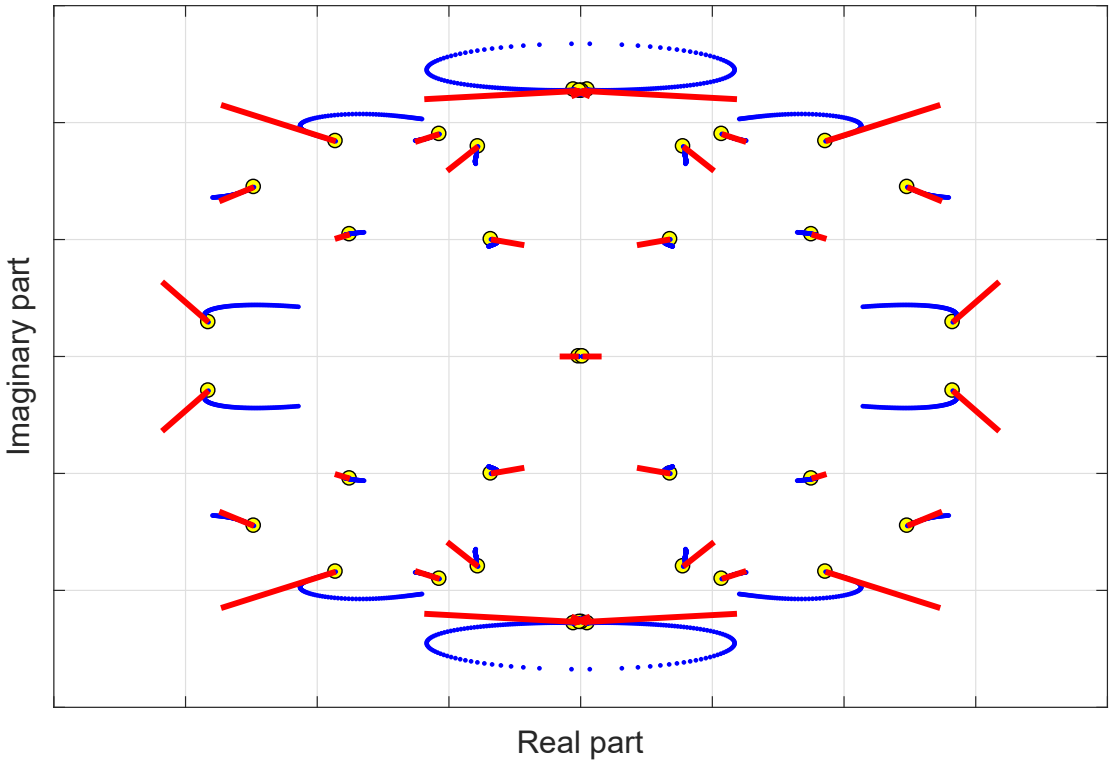
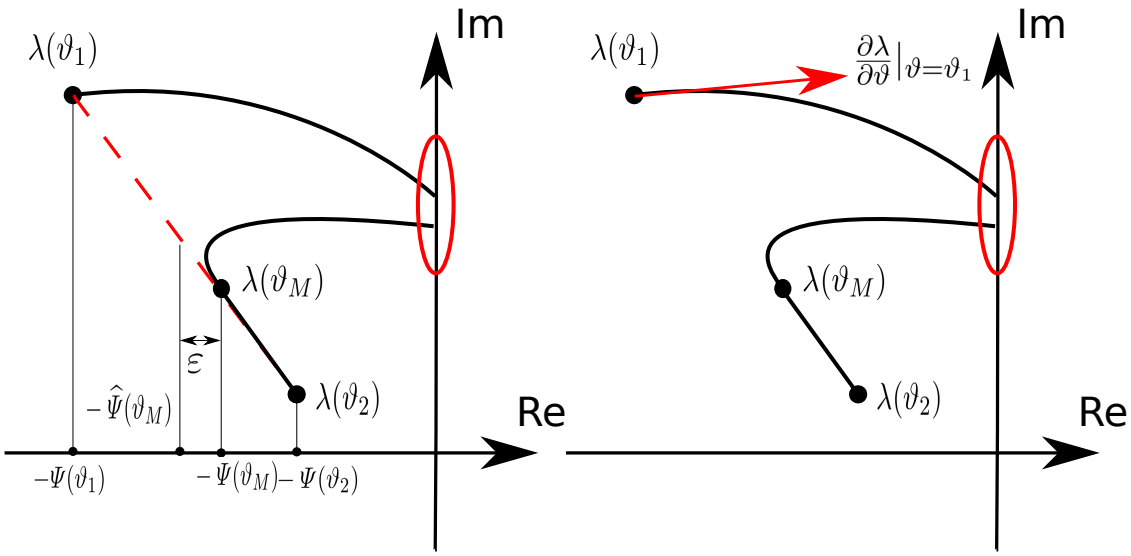


Figure 5.2: Linearly perturbed eigenvalues compared to the exact ones

Summarizing, in this section a first order approximation approach that allows to predict, for a sufficiently small parameter interval, the trajectories of Hamiltonian eigenvalues has been presented. This method is the theoretical background of the adaptive sampling framework that will be discussed in next sections, since it enables to gather information about Hamiltonian eigenvalues without explicitly computing them, thus saving computational time.

5.2 Eigenvalue Perturbation Based Adaptive Sampling Algorithm

As mentioned in the introduction to this chapter, in this section we are going to present a parametric passivity verification algorithm based on Hamiltonian eigenvalue perturbations. Before going into the details of this approach, we briefly discuss how the presented method, from now on denoted as *Derivative-based*, can overcome the problems affecting the Ψ-based scheme. Consider the case illustrated in Figures 5.3:



(a) Ψ-based method (b) Derivative-based method

Figure 5.3: Graphical comparison between the two methods

Both panels show a Hamiltonian eigenvalue trajectory that has high sensitivity with respect to the parameter at $\vartheta = \vartheta_1$ and low sensitivity at $\vartheta = \vartheta_2$. Figure 5.3a shows what would happen using the Ψ-based algorithm: we see that the error between the computed eigenvalue at $\vartheta_M = \frac{\vartheta_1+\vartheta_2}{2}$ and the one predicted with a linear interpolation is small, which means that the sub-interval $[\vartheta_L,\ \vartheta_R]$ is not going to be refined. This leads to losing the violation area highlighted in red.
The new approach we are going to describe relies, instead, on a first order approximation of the true eigenvalue trajectory, as shown in Figure 5.3b. In this way, we are able to detect the eigenvalue high sensitivity in $\vartheta = \vartheta_1$ and refine accordingly the sub-interval, finding the non-passive area.
Additionally, while the Ψ-based algorithm takes into account just the eigenvalues that are closest to the imaginary axis, this new approach allows to keep track of all the Hamiltonian eigenvalues, leading to have a higher sensitivity toward those eigenvalues that, even if far from the imaginary axis, may be displaced close to, or on it.
The algorithm presented here is based again on a coarse initial sampling, at a low refinement level, and on successive adaptive refined passes, if needed.
For us, when a parameter space point is sampled some useful information about the

model, when evaluated at that parameter value, is stored in memory. The sampling process consists mainly in determining passivity violations, if any, with a Hamiltonian driven method and in retrieving all the data necessary to formulate passivity constraints. Moreover, we are interested also in collecting left and right Hamiltonian eigenvectors, that are used, if needed, in eigenvalues perturbations.
All these data are listed below:

- Hamiltonian eigenvalues (used to detect passivity violations as in the non-parameterized case) and their left/right eigenvectors, used in performing eigenvalue perturbations, if needed, as in Equations (5.2) and (5.3);
- Passivity locations, that are stored as frequency-parameter tuples $\{\hat{\omega}, \hat{\vartheta}\}$ corresponding to the largest violation in each considered non-passive sub-band;
- Largest singular value and left/right singular vectors of $\mathbf{H}(\mathrm{j}\hat{\omega}; \hat{\vartheta})$ for scattering systems or smallest eigenvalue and its right eigenvectors of $\mathbf{H}(\mathrm{j}\hat{\omega}; \hat{\vartheta}) + \mathbf{H}^{\mathsf{H}}(\mathrm{j}\hat{\omega}; \hat{\vartheta})$ for immittance systems. These quantities, in addition to the passivity locations, are used in the parametric passivity enforcement to formulate passivity constraints.

The data structure in which all this information is stored is denoted here as $\mathcal{W}$.

5.2.1 Coarse Sampling

In the initial coarse sampling just some of the $2^{j_{max}}$ parameter space points are sampled, according to the initial refinement level j_0. We define as j_{max} the maximum refinement level allowed by the proposed sampling, so that the minimum spacing between two points will be at most $2^{-j_{max}}$. For details on this hierarchical approach see Section 5.3.1. The number of initial points is retrieved from the same heuristic rule as in Section 4.3.2 but, to be compliant with the binary hierarchical grid we use, it must be a power of two. In general, the number of initial samples can not be expressed as a power of two with integer exponent thus, for this reason, and to avoid losing any violation area due to a too coarse sampling, the initial refinement level, computed as the 2-logarithm of the minimum initial samples number estimated as in (4.4), is rounded to the nearest greater integer, as Equation (5.18) shows:

$$j_0 = \lceil \log_2(\kappa \bar{\ell}) \rceil \tag{5.18}$$

where κ is a positive integer and $\lceil \cdot \rceil$ is the ceiling operator.
At this early stage, the algorithm tries to find macro-areas (in the parameter space) in which the identified model is not passive using a Hamiltonian driven scheme. The retrieved information will be used, if necessary, for next refined sweeps. Algorithm 5.6 describes this initial coarse sweep.
Figure 5.4 shows the parameter space after the initial coarse sampling. Circular markers represent the sampled points, our baseline for refined sweeps.

5.2.2 Refined Sweeps

At this stage, if through the coarse sampling we are not able to completely characterize the passivity throughout the parameter space, we proceed with a number of adaptive

Algorithm 5.6 Derivative based coarse sweep

Require: Integer number κ, the parameter basis order $\bar{\ell}$ and the model $\mathbf{H}(s;\vartheta)$

1: Get initial refinement level as in (5.18)
2: **for** $k = 0, \ldots, 2^{j_0}$ **do**
3: Compute the current parameter value $\vartheta_k = k \cdot 2^{-j_0}$
4: Evaluate $\mathbf{H}(s;\vartheta)$ in ϑ_k
5: Get the Hamiltonian matrix associated with $\mathbf{H}(s;\vartheta_k)$
6: Sample the current point as detailed in Section 5.2
7: Update data-structure $\mathcal{W}$
8: **end for**

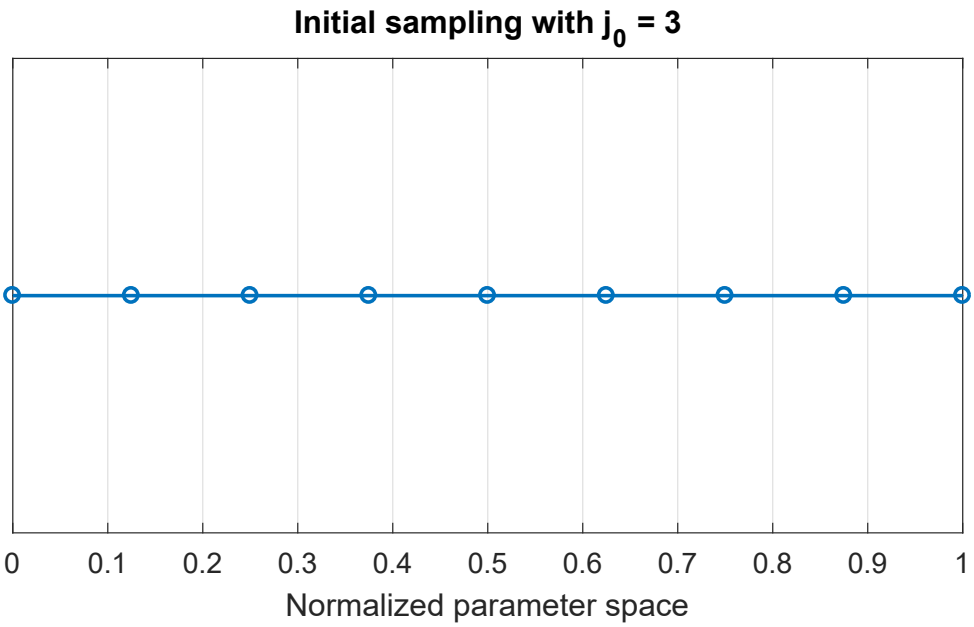


Figure 5.4: Initial sampling at coarse level $j_0 = 3$

refined passes. To this end, the algorithm starts from refinement level j_0 and, for each sub-interval defined by the initial coarse sampling, place a fictitious node, indicated here as 'testing node', between two already sampled points. To make notation simpler, we will use these support variables for readability:

- ϑ_T represent the testing node
- ϑ_L represent the node on the left of ϑ_T
- ϑ_R represent the node on the right of ϑ_T

The testing node is, in general, placed in the middle between ϑ_L and ϑ_R, thus:

$$\vartheta_T = \frac{\vartheta_R + \vartheta_L}{2} \tag{5.19}$$

We must now determine whether the testing node corresponds to a critical parameter value (a possibly passivity violation is close to ϑ_T) or not through first order Hamiltonian eigenvalues perturbations. The framework we are working on is illustrated in Figure 5.5.

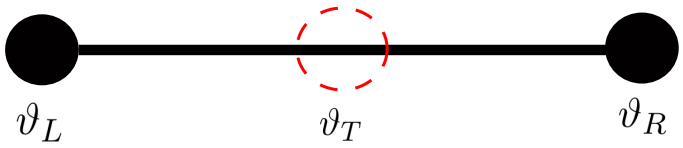


Figure 5.5: Left,right and testing node representation

In details, starting from nodes ϑ_L, ϑ_R, we must compute Hamiltonian eigenvalues perturbations respectively toward ϑ_R, ϑ_L, and checks if some of the perturbed eigenvalues become critical. For us, an Hamiltonian eigenvalue is critical if, after a small parameter variation, it may become purely imaginary, identifying the presence of a non-passive area. This double check (left to right and vice-versa) is required due to Hamiltonian eigenvalues non-linearities. If the the first perturbation detects a critical area, the other direction is not tested since it would be useless and no additional information would be retrieved.
To better explain this scheme, Figure 5.6 shows these two perturbations in parameter space sub-interval, from two edge nodes ϑ_L, ϑ_R toward the testing node ϑ_T.

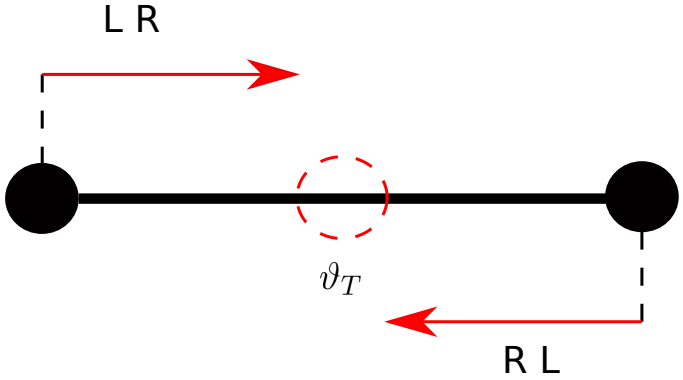


Figure 5.6: Left-Right and Right-Left perturbation graphical representation

We now formalize the above scheme. Applying the perturbation operator defined in Equation (5.17) to nodes ϑ_L and ϑ_R, and with the parameter variation equal to:

$$|\delta\vartheta| = \frac{1}{2}(\vartheta_R - \vartheta_L) \tag{5.20}$$

we have:

$$\begin{cases} \widehat{\Lambda}_{\vartheta_L}(\vartheta_T) = \mathcal{P}_{M,K}^{|\delta\vartheta|}(\vartheta_L) \\ \\ \widehat{\Lambda}_{\vartheta_R}(\vartheta_T) = \mathcal{P}_{M,K}^{-|\delta\vartheta|}(\vartheta_R) \end{cases} \tag{5.21}$$

where the first refers to the left-to-right perturbation and the second to the right-to-left one.

Hamiltonian eigenvalues trajectories may be, in some cases, strongly non-linear and a perturbation toward ϑ_T may be not enough to detect a violation. For this reason, in the algorithm implementation we leave as a degree of freedom the perturbation extent. This is made by defining a quantity, called α, whose value is related to the perturbation extent by (5.22):

$$\alpha = \frac{\vartheta - \vartheta_L}{\vartheta_T - \vartheta_L} \tag{5.22}$$

or, equivalently, by

$$\alpha = \frac{\vartheta_R - \vartheta}{\vartheta_R - \vartheta_T}$$

where ϑ denotes the perturbation end point.
Thus, indicating with $\delta\vartheta^*$ the new perturbation amplitude, it holds that

$$|\delta\vartheta^*| = \alpha \ \ |\delta\vartheta| \tag{5.23}$$

In most of the cases it has been noted that a value $\alpha = 1$ is enough to guarantee the algorithm to find all the violations but, to ensure its functioning under any working condition, this value has been set to $\alpha = 2$, making it more reliable at the cost of a slightly higher computational effort.

We are now ready to discuss in details the rules upon which the algorithm adaptively samples the parameter space.
As theoretically derived in Section 1.6.2, the Hamiltonian eigen-spectrum symmetry justifies the choice to consider, without loss of generality, just the 2^{nd} quadrant eigenvalues ($\mathrm{Re}\{\lambda\} \leq 0$ and $\mathrm{Im}\{\lambda\} \geq 0$) with related perturbations.
In this framework, recalling that an eigenvalue is defined to be critical if there is a high probability that it becomes imaginary due to a small parameter variation, we mainly focus on two metrics: the eigenvalue trajectory direction and its distance from the imaginary axis. Quantitative information about the trajectory direction can be retrieved from (5.21) and, to decide if an eigenvalue is too close to the imaginary axis we introduce a threshold $\delta_{th} \geq 0$.
How these metrics are used to determine if a parameter space area is critical or not is detailed below:

1. If the real part of at least one 2^{nd} quadrant perturbed Hamiltonian eigenvalue is greater than $-\delta_{th}$ and its trajectory points toward the imaginary axis, it is likely that a critical area is opening and it must be sampled. In the following we will refer to this case as "*Case 1*". See Figure 5.7a;

2. If the real part of a 2^{nd} quadrant eigenvalue is less than $-\delta_{th}$ we check if its real part perturbation is greater than $-\delta_{th}$: if this happens there is a high probability that

with a small parameter variation we move toward a non-passive area, so additional samples must be added. In the following we will refer to this case as "*Case 2*". See Figure 5.7b;

3. If the real part of a 2^{nd} quadrant Hamiltonian eigenvalue is less than $-\delta_{th}$ and it is moving away from the imaginary axis, then for a sufficiently small parameter variation no passivity violations are likely to be found. In this case no additional samples are required. In the following we will refer to this case as "*Case 3*". See Figure 5.7c;

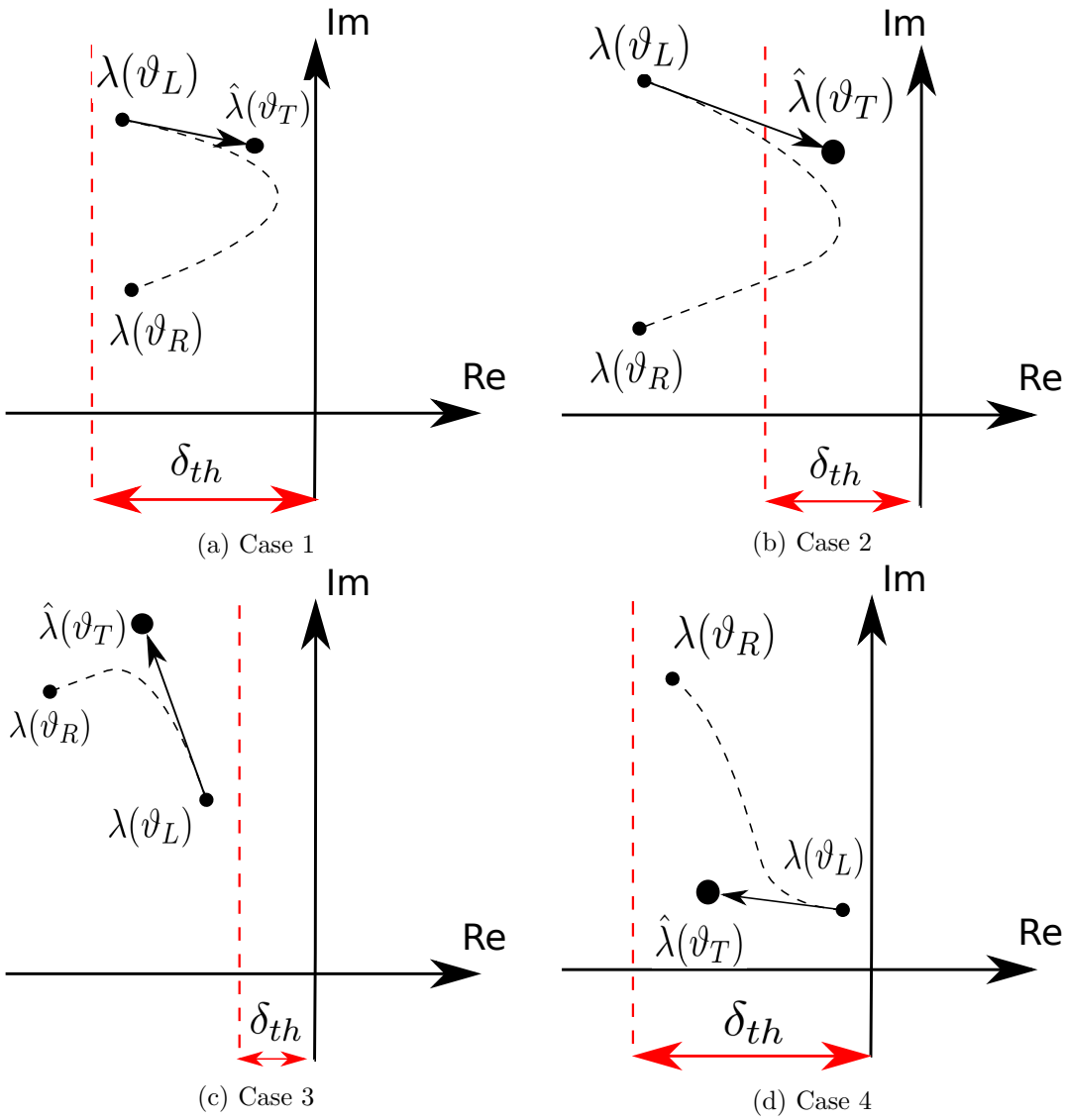
4. If a 2^{nd} quadrant Hamiltonian eigenvalue real part is greater than $-\delta_{th}$ but it is moving away from the imaginary axis, no additional samples are required. This condition consider the case in which a violation area has just disappeared (some eigenvalues may be still very close to the imaginary axis without being necessarily critical), and avoid to finely sample non-critical regions. In the following we will refer to this case as "*Case 4*". See Figure 5.7d.

These rules are reported in Table 5.1 (ϑ_0 is intended to be both ϑ_R and ϑ_L depending on the perturbation direction). The term λ_{ϑ_0} represents a 2^{nd} quadrant Hamiltonian eigenvalue evaluated at ϑ_0, $\frac{\partial\lambda(\vartheta)}{\partial\vartheta}|_{\vartheta=\vartheta_0}$ represents its derivative with respect to the parameter and $\widehat{\lambda}_{\vartheta_0}(\vartheta_0+\delta\vartheta)$ is its perturbation, for a parameter variation equal to $\delta\vartheta$.

Case	$\mathrm{Re}\{\lambda_{\vartheta_0}\} < -\delta_{th}$	$\mathrm{Re}\left\{\frac{\partial\lambda(\vartheta)}{\partial\vartheta}|_{\vartheta=\vartheta_0}\right\} > 0$	Critical ?
1	NO	YES	YES
2	YES	YES	Check $\mathrm{Re}\left\{\widehat{\lambda}_{\vartheta_0}(\vartheta_0+\delta\vartheta)\right\}$
3	YES	NO	NO
4	NO	NO	NO

Table 5.1: Mono-variate critical cases for adaptive refinement

By looking at the previous table, we see that just the first two cases (namely, Case 1 and Case 2) are critical.
Furthermore, before performing this refined verification, the algorithm is set to execute a preliminary test that mainly serves to avoid refinements in already non-passive defined areas. It consists in detecting, through a Hamiltonian-driven test, if in both the sub-interval edges ϑ_L and ϑ_R the model is not passive and, if true, no additional samples are required to better characterize these zones that are considered as not-passive. In Figure 5.8a we show this case, where the red dots represent the non-passive sub-interval edges. Additionally, if one of the two edges is passive and the other is not, no refined verification are in order to completely characterize this sub-interval, that is critical (a passivity violation area opens or closes passing from ϑ_L to ϑ_R) and it must be refined. Figure 5.8b shows this case, where the red dot is the non-passive edge while the green dot represent the passive one.
This preliminary test enables then to avoid refined tests on areas where it is not required and, in some cases, to directly refine a sub-interval without performing any eigenvalue perturbation. Algorithm 5.7 formulates the proposed adaptive sampling scheme



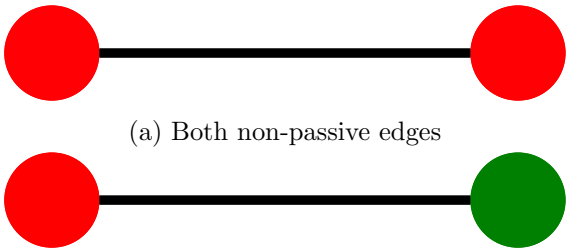
(a) Case 1 (b) Case 2

(c) Case 3 (d) Case 4

Figure 5.7: Graphical representation of refined sweeps cases

5.3 Practical Implementation

In this section a number of practical implementation issues are discussed. The adaptive sampling scheme, shown in Section 5.2, if implemented as is, may fail in detecting some violation areas. This is due to the fact that not all the issues related to parametric passivity verification, such as asymptotic singular values behaviour and parameter space tessellation self-consistency, have been addressed.



(a) Both non-passive edges

(b) Passive/non passive edges

Figure 5.8: Graphical representation of the preliminary test

In the next sections we go through these problems, proposing a solution for each of them, with detailed algorithms and numerical results.

5.3.1 Hierarchical Approach

Motivation

While performing the adaptive sampling it is important to give a well defined structure to the parameter space, leading to a better performances of the whole algorithm. For this reason, it has been chosen to adopt a hierarchical approach, which means that all the points on our parameter space belongs to a tree structure at a certain hierarchical level.
The main driving factor to adopt this kind of structure is numerical stability. In fact, this approach enables the use of only integers instead of floating point numbers to define a certain point in the parameter space, eliminating then the possibility of truncation errors. This particular structure makes it easy to improve the algorithm also from other standpoints. For example, it is possible to make our sampled parameter space self-consistent, allowing the sampling to be more effective, as explained in the related section.

Implementation

To better explain this hierarchical approach it is necessary to normalize the parameter space. To this end, we take the usual parameter space, 1-D for simplicity, defined as $[\vartheta_{min},\ \vartheta_{max}]$, and normalize it to a $[0,\ 1]$ segment. This is possible through this simple linear transformation:

$$\tilde{\vartheta} = \frac{\vartheta - \vartheta_{min}}{\vartheta_{max} - \vartheta_{min}}$$

where $\tilde{\vartheta}$ is the normalized parameter value.
Considering then the normalized parameter-frequency space, it is possible to map set of points on a uniform grid with spacing 2^{-j} to a tuple of positive integers $\{k,\ j\}$ through this relation:

$$\tilde{\vartheta} = k \cdot 2^{-j}$$

More specifically, index j represents the refinement level, expressed as the negative exponent of 2, since a bisection approach is used. The other index $k \in \{0,1,...,2^j\}$ is a linear

Algorithm 5.7 Derivative-based adaptive sampling

Require: Parametric model $\mathbf{H}(s;\vartheta)$
Require: Perturbation amplification factor α as defined in (5.22)
for $j = j_0, \ldots, j_{max}$ **do**
  Find k-indices of points at current refinement level $\{k_1, \ldots, k_m\}$
  for $k = \{k_1, \ldots, k_m\}$ **do**
    Compute $\vartheta_L = (k-1)\cdot 2^{-j}$ and $\vartheta_R = (k+1)\cdot 2^{-j}$
    Define ϑ_T as in Equation (5.19)
    Get Hamiltonian pencils $(\mathbf{M}, \mathbf{K})$ associated with $\mathbf{H}(s;\vartheta_L)$ and $\mathbf{H}(s;\vartheta_R)$
    Check if $\mathbf{H}(s;\vartheta_L)$ and $\mathbf{H}(s;\vartheta_R)$ are passive.

    if $\mathbf{H}(s;\vartheta_L)$ and $\mathbf{H}(s;\vartheta_L)$ are both non-passive **then**
      break
    else
      Sample ϑ_T as in Section 5.2
      Update data-structure $\mathcal{W}$
      break
    end if

    Define $\delta\vartheta$ as in (5.20) and $\delta\vartheta^*$ as in (5.23)
    Get perturbed eigen-spectra $\widehat{\Lambda}_{\vartheta_L}(\vartheta_L + |\delta\vartheta^*|) = \mathcal{P}_{M,K}^{|\delta\vartheta^*|}(\vartheta_L)$, $\widehat{\Lambda}_{\vartheta_R}(\vartheta_R - |\delta\vartheta^*|) = \mathcal{P}_{M,K}^{-|\delta\vartheta^*|}(\vartheta_R)$

    if Case 1 critical **then**
      Sample ϑ_T as in Section 5.2
      Update data-structure $\mathcal{W}$
      break
    else if Case 2 critical **then**
      Sample ϑ_T as in Section 5.2
      Update data-structure $\mathcal{W}$
    end if

  end for
end for

index that identifies a specific point on a uniform grid at the j-th refinement level.
To better explain how a hierarchical structure is built, Figure 5.9 (defined as "*ladder*" representation of the parameter space) compares the usual linear partitioning of the parameter space with respect to the hierarchical one.

This mapping is not injective, since each point $\vartheta \in [0,\ 1]$ may be represented by an infinite number of tuples $\{k,\ j\}$ just by taking a suitable index k for a given index j.
Example:
Taking $\tilde{\vartheta} = \frac{1}{4}$, it is easy to show that it corresponds to indices $k^* = 1,\ j^* = 2$.

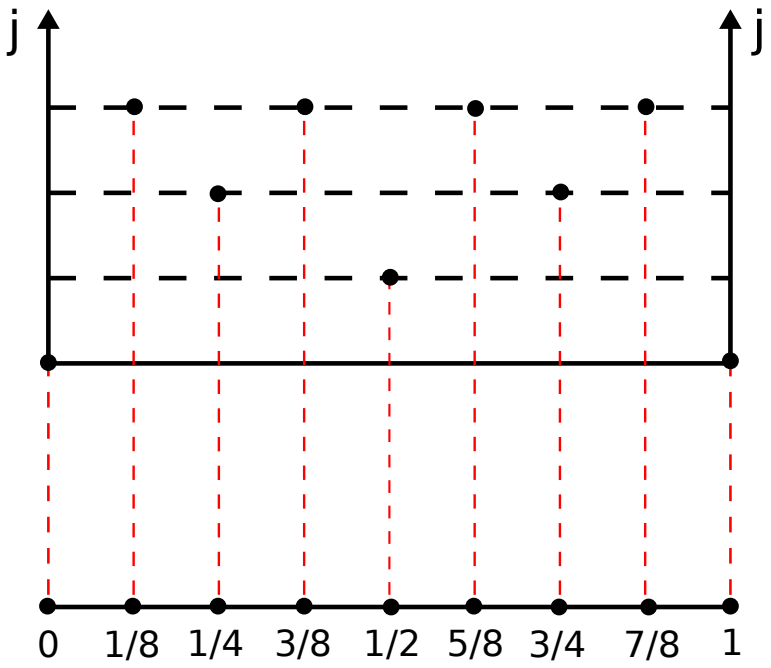


Figure 5.9: Linear space representation compared to hierarchical

However, the same normalized parameter $\tilde{\vartheta}$ correspond to hierarchical indices $k_1 = 2$, $j_1 = 3$ and, in general, to any tuple $\{k_i,\ j_i\}$ with $k_i = k^* \cdot 2^i$, $j_i = j^* + i$, for $i \in \mathbb{N}$.

For our purposes this may induce ambiguities in defining points along the normalized parameter axis. On the other hand, we can overcome this problem ensuring that, for any tuple $\{k,\ j\}$, k-index is always odd. Indices that are guaranteed to satisfy this condition are expressed in what will be called the "*reduced form*".
To ensure that only reduced indices are used, an index-reduction routine has been implemented. Starting from any tuple $\{k,\ j\}$, it returns the reduced indices by halving k and correspondingly decreasing j, until k is odd.
Algorithm 5.8 represent this procedure.

5.3.2 Grid Self-Consistency

Motivation

The proposed bisection approach, used inside the adaptive sampling method, is very effective in precisely locating passivity violations. However it has a major drawback. In fact, this refinement strategy, without any control, may lead to have abrupt refinement levels changes on adjacent parameter space subsets. This mainly happens near passive/non-passive interface regions where, moreover, it is important to detect all passivity violations.

Algorithm 5.8 Index reduction algorithm

Require: $k,\ j$ indices
1: **if** $k = 0$ **then**
2: $\quad j = 0$
3: **return** $\{k, j\}$
4: **end if**
5: **if** $k = 2^j$ **then**
6: $\quad j = 0$
7: $\quad k = 1$
8: **return** $\{k, j\}$
9: **end if**
10: **while** k is even **do**
11: $\quad k \leftarrow \frac{k}{2}$
12: $\quad j \leftarrow j - 1$
13: **end while**
14: **return** $\{k, j\}$

The algorithm, as is, can not smoothly check in these areas, leading to a reduced accuracy.
As anticipated, the particular structure we gave to the parameter space lead to easily improve the passivity check algorithms. In this scope, a parameter space hierarchical structure allows to easily overcome this problem, because it enables to precisely place additional samples on the edges of the above discussed critical areas, thus enabling more reliable passivity verifications.

Implementation

A parameter-space grid is defined to be self-consistent if it does not contain steep variations on refinement levels on adjacent samples. In details, for us self-consistency means that the maximum difference on refinement levels between two adjacent samples can not be greater than one. In order to fulfill this condition, an a-posteriori algorithm, defined as "*grid-fixing algorithm*", samples the required points to make the grid self-consistent.
Denoting with j_0 and j_{max} the lowest and highest refinement levels respectively, the proposed grid-fixing algorithm checks if for each point $\hat{k}$ at refinement level $\hat{\jmath}$, the left and right adjacent samples exist on the level $\hat{\jmath} - 1$. If a point is missing, it is sampled and added to the data-structure.
In details, considering a normalized parameter point identified by its hierachical indices $\{\hat{k},\ \hat{\jmath}\}$, we define its right and left adjacent samples with the tuples:

$$\{k_R,\ j_R\} = \left\{ \left[\!\left[\frac{\hat{k}+1}{2} \right]\!\right],\ \hat{\jmath} - 1 \right\} \tag{5.24}$$
$$\{k_L,\ j_L\} = \left\{ \left\lfloor \frac{\hat{k}-1}{2} \right\rfloor,\ \hat{\jmath} - 1 \right\}$$

where $\{k_R,\ j_R\}$ and $\{k_L,\ j_L\}$ are respectively right and left adjacent samples indices and the operators $\lceil\!\lceil \cdot \rceil\!\rceil$, $\lfloor\!\lfloor \cdot \rfloor\!\rfloor$ round their arguments to the nearest greater and smaller odd integers, respectively

A decreasing-level strategy (from j_{max} to j_0) is adopted because it does not need any iteration to accomplish a complete grid-fixing, as would be required by an increasing-level (from j_0 to j_{max}) approach.
A sketch of this procedure is shown in Algorithm 5.9.

Algorithm 5.9 Grid-fixing algorithm

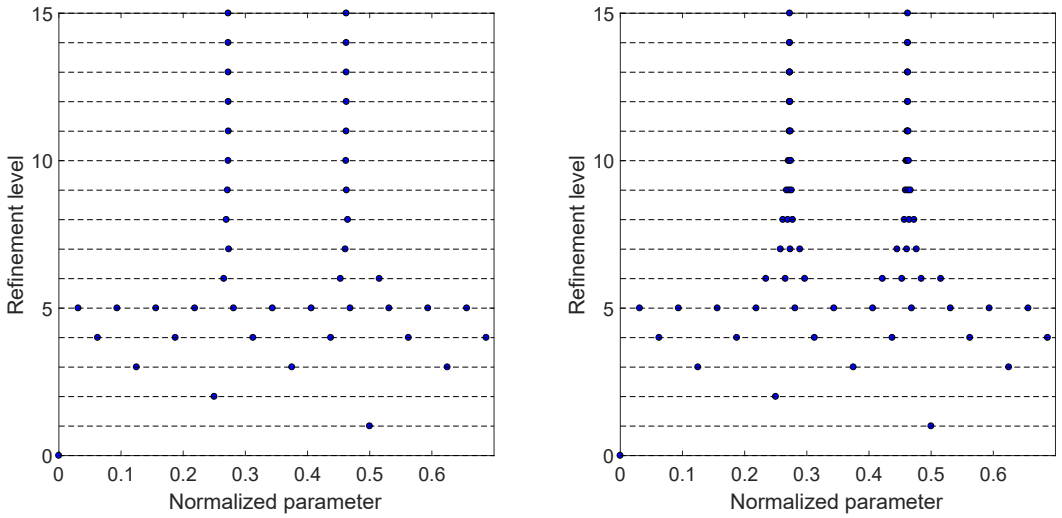
```
for ĵ = [j_max, −1, j_0] do
    for k = 1, 3, 5, ..., 2^ĵ − 1 do
        Get left {k_L, j_L} and right {k_R, j_R} samples indices as in (5.24)
        Define ϑ_L = k_L · 2^(−j_L) and ϑ_R = k_R · 2^(−j_R)
        if ϑ_L or ϑ_R ∉ W then
            Sample missing parameter points as detailed in Section 5.2
            Update data-structure W
        end if
    end for
end for
```

This algorithm, as is, may wrongly sample points outside the normalized parameter space $[0,\ 1]$, when trying to "fix" boundary areas that are, not belonging to the normalized grid, not self-consistent with it by definition. In order to fix this problem the final algorithm has some additional controls on the values to be added.

In the following we present a numerical example where, through a graphical interpretation, we show the difference between an initially non-self consistent grid and the same grid made self-consistent by the proposed algorithm.
In particular, Figure 5.10a shows, in the ladder representation, the non self-consistent grid while Figure 5.10b shows the self-consistent one. Looking at the right panel, it is clear that each grid point has on its right and on its left the corresponding samples at the previous refinement level while, the one on the left, does not satisfy our self-consistency definition.

5.4 Numerical Results

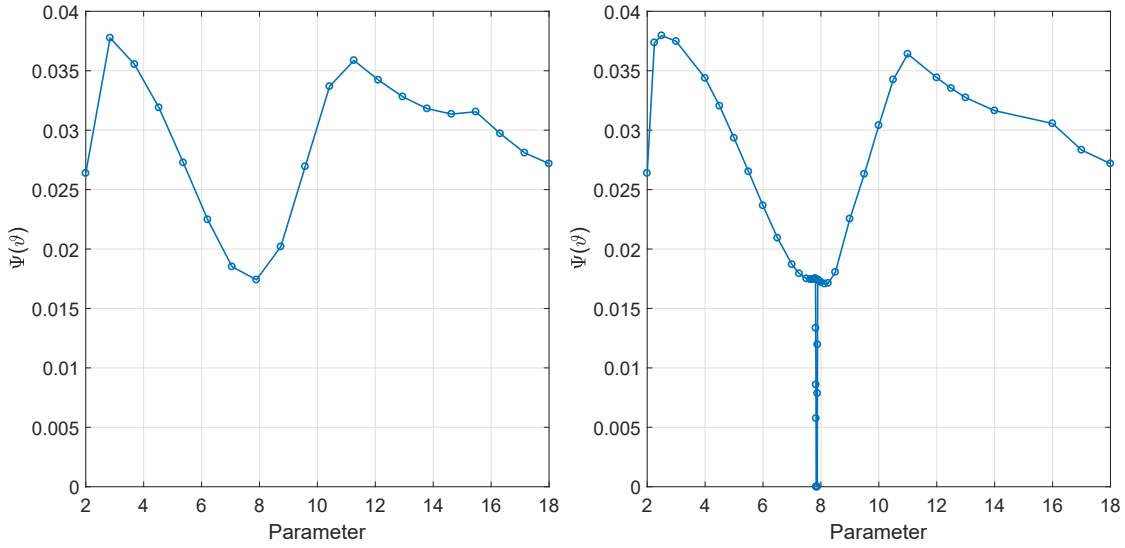
In this section we are going to provide several numerical results that compare the Ψ-based approach with the one presented in this chapter.
At the beginning of this chapter we showed that the Ψ-based algorithm failed in detecting a non-passive area.
Here, we propose the same test-case by comparing the $\Psi(\vartheta)$ functions resulting from the



(a) Non self-consistent grid (b) Self-consistent grid

Figure 5.10: Non self-consistent VS self-consistent grid (*Test Case 2–b*)

two passivity verification approaches. Figure 5.11 shows these results. It is clear from



(a) Ψ-based method (b) Derivative-based method

Figure 5.11: Passivity verification methods comparison (*Test Case 2–b*)

the right panel, showing the outcome of the derivative-based verification algorithm, that the new approach is able to spot a very small non-passive region, the one we saw in the introductory section through the fine singular values sampling, while the Ψ-based, whose result is shown in the left panel, does not. This performance improvement is mainly due

to the predictive aspects of eigenvalue derivatives: in fact, by looking at the Hamiltonian eigenvalues real part as a function of the parameter in Figure 5.12, we see that there are two Hamiltonian eigenvalues that move fast toward the imaginary axis, become imaginary for a small parameter interval, then return to be complex. The derivative based method detects this phenomenon by checking the real part of eigenvalues derivatives, while the Ψ-based one is effective just if the initial coarse sampling samples a point in this small interval.

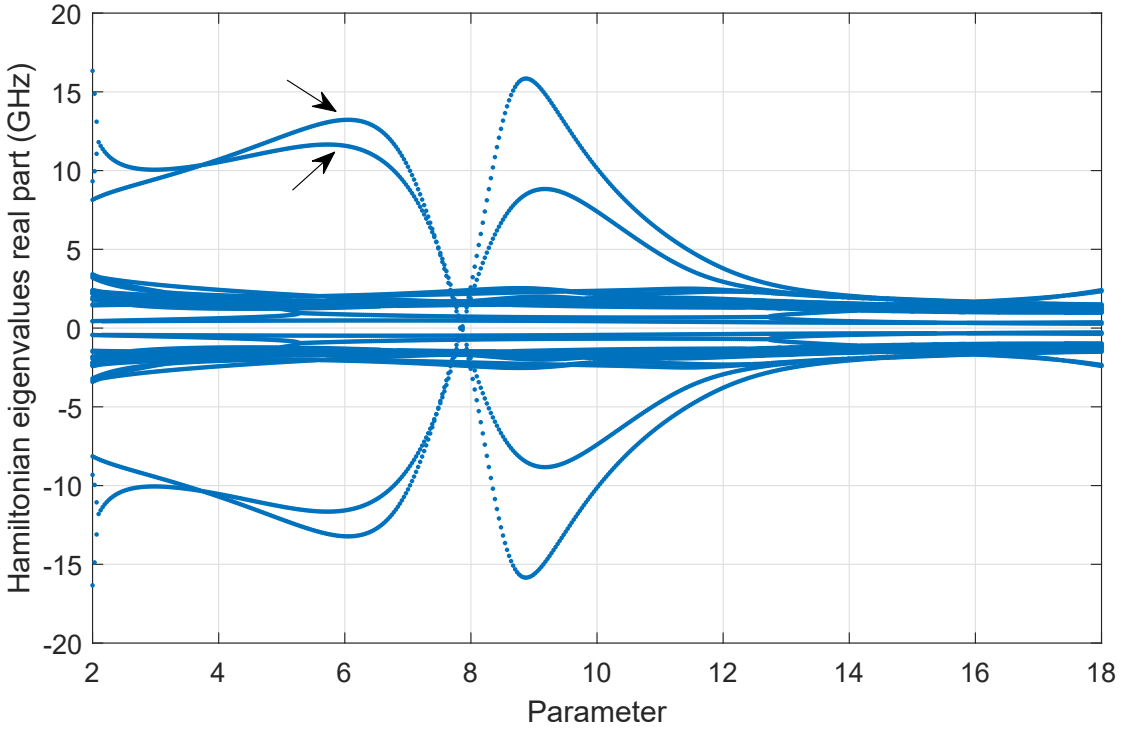


Figure 5.12: Candidate passive model Hamiltonian eigenvalues real part (*Test Case 2–b*)

Figure 5.13 shows the same phenomenon from the complex plane perspective. The blue crosses represent the Hamiltonian eigenvalues evaluated for $\vartheta = 7.7$ that are falling rapidly towards the imaginary axis. These eigenvalues become purely imaginary for $\vartheta \approx 7.84$ (red crosses) and come back to have a non-vanishing real part for $\vartheta \approx 7.85$ (green crosses). Finally they move away from imaginary axis toward yellow crosses for $\vartheta \approx 8.1$. This is not an isolated case. In fact, due to the high non-linearity of Hamiltonian eigenvalues, these kind of trajectories are not uncommon. For the sake of completeness, Figures 5.14a and 5.14b compare the two verification methods on another test-case (*Test Case 12–b*). As in the previous one we have a sharp variation of $\Psi(\vartheta)$ that it is not detected by the Ψ-based algorithm. The derivative-based one is able to locate it detecting the presence of some eigenvalues that have a high sensitivity to the parameter in this area.

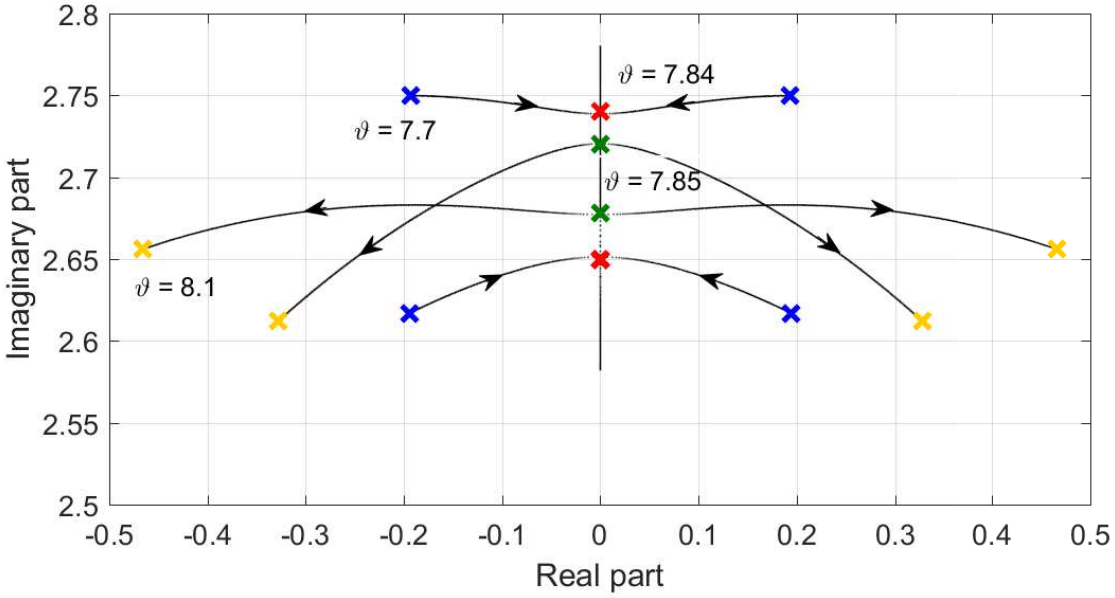
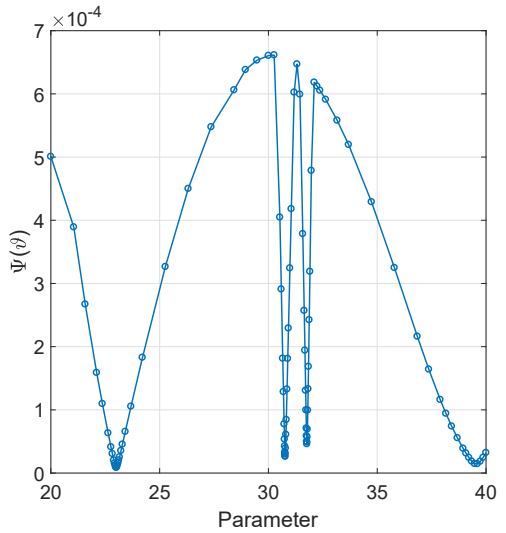
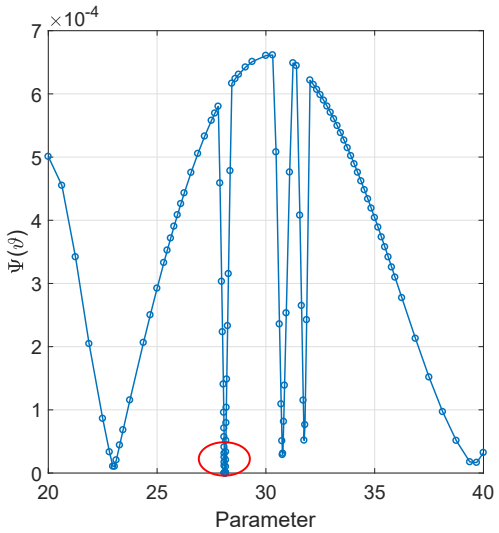


Figure 5.13: Candidate passive model eigenvalues trajectories (*Test Case 2–b*)



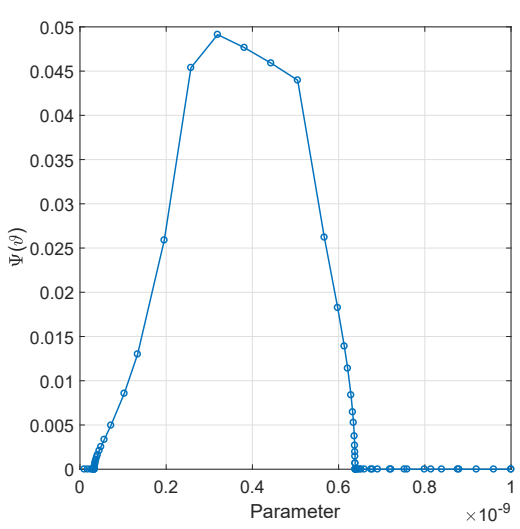
(a) Ψ-based passivity verification result (*Test Case 12–b*)



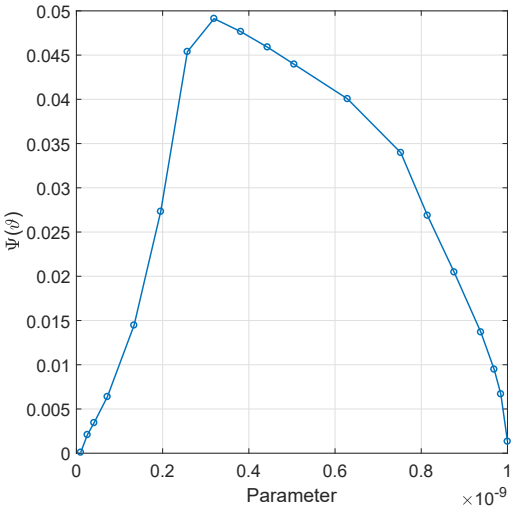
(b) Derivative-based passivity verification result (*Test Case 12–b*)

In the previous examples we saw that the new approach is more reliable in detecting small violations induced by quick Hamiltonian eigenvalues variations.

We are going now to exploit the proposed verification scheme to enforce passivity by means of the enforcement procedure discussed in Section 4.4. To this end, starting from a non-passive model (for the structure see *Test Case 6* in Appendix A), whose $\Psi(\vartheta)$ function is reported in 5.15a, we try to enforce it to be passive. We see that the enforcement algorithm, combined with the derivative-based verification method, is capable of returning a passive model, whose $\Psi(\vartheta)$ function is reported in 5.15b, in 4 iterations, with a total CPU time of 6.63 seconds.



(a) Non-passive model $\Psi(\vartheta)$ function



(b) Passive model $\Psi(\vartheta)$ function (fourth iteration)

Figure 5.15: Comparison on $\Psi(\vartheta)$ functions (*Test Case 6*)

About the passive model accuracy, Figure 5.16 compares the scattering responses of the passive model with respect to data. We notice that the passive model is still very accurate: quantitatively, the maximum relative error between the passive model response and data is $2.5 \cdot 10^{-3}$, almost equal to the one of the non-perturbed model. These results show that the combination of the derivative-based adaptive sampling with the parametric enforcement algorithm, detailed in Section 4.4, is capable of reliably enforcing passivity on parametric macromodels without affecting the accuracy.

A key point for passivity verification algorithms are computational times, since they are used repeatedly during passivity enforcement procedures. Thus, even if the derivative based algorithm has proved to be more reliable, we must ensure that it can be used to accomplish in reasonable time a complete passivity enforcement. To this end, in the following, we present a comparison between the two passivity verification approaches, mainly in terms of:

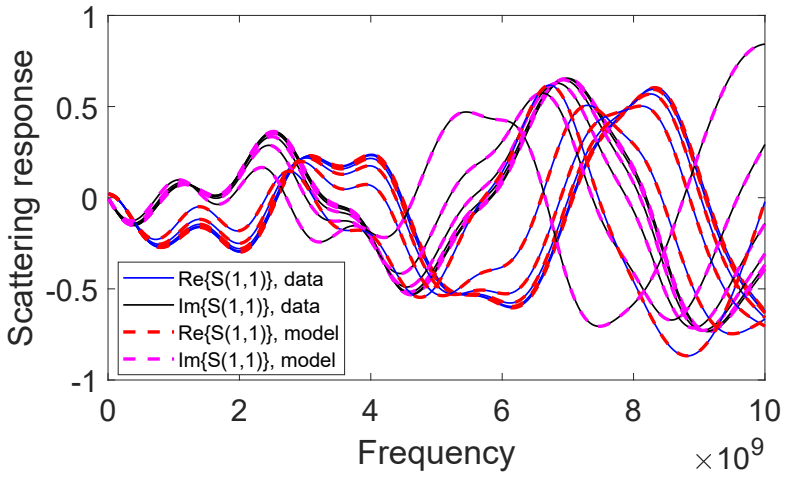


Figure 5.16: Comparison between model response and data (*Test Case 6*)

- Passivity enforcement computation time;
- Passivity enforcement number of iterations;
- Passivity enforcement outcome (0: non passive, 1: passive);
- A cross-check indicating if a model, defined passive for a method, is still passive for the other (0: non passive, 1: passive).

The settings used for the Ψ-based algorithm are:

- Maximum number of successive refinements = 10;
- Linear interpolation error threshold = 0.2;
- Additional singular values/eigenvalues = 30;
- Non-self-consistent grid.

while for the derivative based are:

- Maximum number of successive refinements = 10;
- Perturbation amplification factor $\alpha = 2$;
- Threshold $\delta_{th} = 10^{-6}$;
- Additional singular values/eigenvalues = 30;
- Self-consistent grid.

Table 5.2 shows a comparison between the two methods for models with ensured positive-real denominator (for details see [22], [8], [9]), from now on denoted as *"PR"*, while Table 5.3 shows the same comparison made this time on models whose denominator is not ensured to be positive-real throughout the parameter space. The latter will be denoted as *"not-PR"*.

Test case	Ψ-based algorithm				Derivative based algorithm			
	Time [s]	Iterations	Outcome	Derivative cross check	Time [s]	Iterations	Outcome	Ψ cross check
1	112	34	1	1	64.2	17	1	1
2–a	298	40	0	0	273	40	0	1
3–a	31.7	6	1	1	35.4	6	1	1
4–a	36.1	4	1	1	48.8	5	1	1
5	31.1	7	1	1	35.3	9	1	1
6	13.3	4	1	1	13.6	4	1	1
7	55.8	7	1	1	25.9	5	1	1
8	0.974	0	1	1	0.739	0	1	1
9	137	26	1	1	47.6	6	1	1
10	391	35	1	1	344	38	1	1
11	138	14	1	1	81.5	12	1	1
12	191	7	1	1	651	6	1	1
13	767	11	1	0	977	12	1	1

Table 5.2: Ψ-based algorithm VS derivative-based algorithm. Test on models with ensured positive real denominator

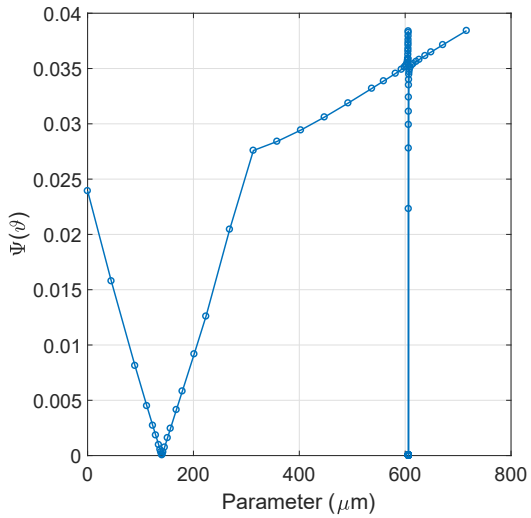
Test case	Ψ-based algorithm				Derivative based algorithm			
	Time [s]	Iterations	Outcome	Derivative cross check	Time [s]	Iterations	Outcome	Ψ cross check
1	40.2	8	1	0	304	9	1	1
2–a	562	40	0	0	660	40	0	0
3–a	36.1	9	1	1	24.8	5	1	1
4–a	96.7	8	1	0	257	27	1	1
5	28.3	7	1	1	16.4	6	1	1
6	7.66	2	1	1	8.73	3	1	1
7	37.3	5	1	1	29.2	5	1	1
8	0.638	0	1	1	0.588	0	1	1
9	298	6	1	1	33.6	6	1	1
10	3602	26	1	1	112	19	1	1
11	87.2	15	1	1	70.0	9	1	1
12	122	6	1	0	157	6	1	1
13	667	11	1	0	1041	8	1	1

Table 5.3: Ψ-based algorithm VS derivative-based algorithm. Test on models without ensured positive real denominator

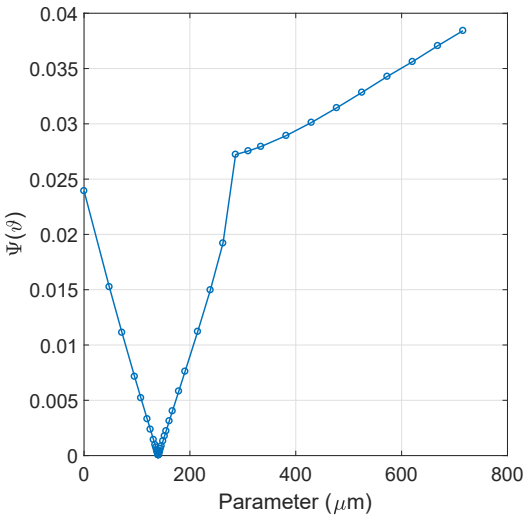
For Test Cases descriptions see Appendix A.

Analyzing these results we see that, in general, the CPU time required to accomplish a complete passivity enforcement is similar for the two approaches. Sometimes, however, the computational effort required by the derivative-based approach is remarkably lower, as in *Test Cases 1, 7, 9, 11* in Table 5.2 and for *Test Cases 9, 10* in Table 5.3.
On the other hand, there are cases for which the proposed sampling scheme is more time-consuming, as in *Test Case 12*. This particular test-structure is characterized by a considerable number of ports (6) thus, computing Hamiltonian eigenvalue perturbations may become a very expensive task, leading to higher computational times.
About the reliability, both Tables confirm the results we detailed previously. In fact, there are several test cases in which the Ψ-based method labels as passive models that, when checked with the derivative-based method, show residual passivity violations for some parameter values. We can notice this fact in *Test Cases 2–a, 13* in Table 5.2 and in *Test Cases 1, 4–a, 12, 13* in Table 5.3.
Test Case 2–a deserve attention, in fact we see that for none of the two models, PR and not-PR, the enforcement algorithm has been capable to return a passive model within 40 iterations. In this case, due to the particular behavior of Hamiltonian eigenvalue trajectories, 49 enforcement iteration were required.

A last consideration must be made about models with not ensured positive-real denominator: these models may present, for some parameter values, a vanishing denominator that makes the whole function not differentiable. This fact, in practice, may cause the eigenvalues derivatives to blow up in amplitude, in turn requiring that unnecessary parameter samples could be added, slowing down the entire passivity enforcement process. This explains why, for some cases in Table 5.3, the time to enforce the passivity is, apparently, unnecessarily high.
To show this phenomenon, we provide an example, based on *Test Case 3–b* (see Appendix A), where the model denominator has a singularity for $\vartheta \approx 600$ μm. In Figure 5.17a, we report the function $\Psi(\vartheta)$ obtained with the derivative-based method: we see that, when ϑ approaches the critical value of 600 μm, the algorithm densely samples the area since derivative values blow up. Furthermore, by comparing Figure 5.17a with Figure 5.17b, the latter representing the function $\Psi(\vartheta)$ obtained with the Ψ-based method, we notice that the Ψ-based approach is not able to detect this violation, labeling as passive the model. This is another evidence that the proposed derivative-based scheme greatly improves the sampling reliability with respect to Ψ-based one.
Anyway, passive or not, a model with a singularity located inside the parameter space is not feasible for simulation. On the other hand, state-of-the-art macro-model extraction algorithms [8, 9, 22], are able to enforce the model stability acting on the model denominator, forcing its real part to be uniformly positive throughout the parameter space. This lead to avoid denominator singularities, overcoming the problems discussed above.



(a) $\Psi(\vartheta)$ obtained with derivative-based approach



(b) $\Psi(\vartheta)$ obtained with Ψ-based approach

Figure 5.17: Denominator singularity effects on $\Psi(\vartheta)$ function (*Test Case 3–b*)

5.5 Parametric Asymptotic Passivity Characterization

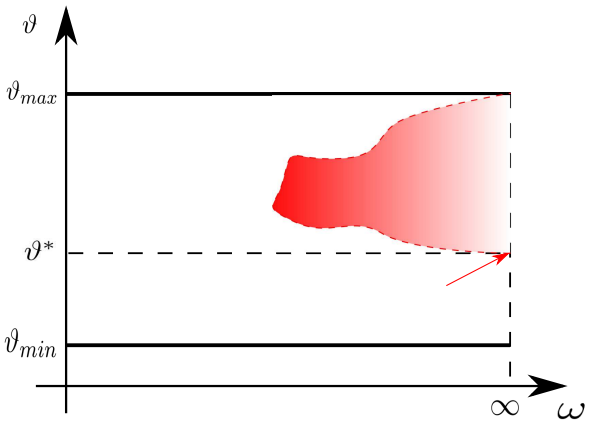
A model is defined asymptotically passive if passivity conditions in Section 1.6.2 are verified when $s = \mathrm{j}\omega \to \infty$. Here, however, we are dealing with parametric macro-models and our aim is to guarantee asymptotic passivity for any feasible value attained by the parameter. In this work, we will discuss about the two main causes that may lead a parametric macro-model to be non-asymptotically passive, that are listed below:

- Vanishing model denominator
  If the model denominator approaches zero, for a bounded input energy, the output one will diverge towards infinity, then there is a passivity violation.

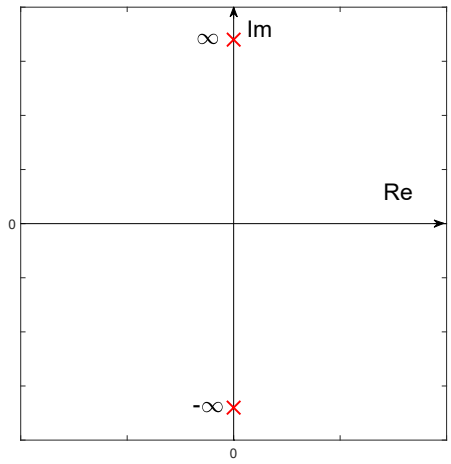
- Model passivity violation
  A more generic situation is where the model is "well-behaved", so that the denominator does not vanishes in the considered parameter space, and asymptotic passivity violations are found for some parameter value.

Instead of looking at the well-known passivity conditions for LTI systems, there is an interesting alternative way to look at asymptotic passivity that is related to the Hamiltonian eigen-spectrum.
From Section 1.6 we know that the purely imaginary eigenvalues of the Hamiltonian matrix associated with the system are located exactly at the frequencies at which passivity violations occur. In our case, since we are focusing on asymptotic violations, these eigenvalues will have infinite imaginary part. Supposing that such a violation occurs for $\vartheta = \vartheta^*$ and that the model is passive $\forall \vartheta \in [\vartheta_{min},\ \vartheta^*)$, we can represent the situation

as in Figures 5.18a and 5.18b. In the left panel we show the violation (red area) in the parameter-frequency plane while the right one shows the Hamiltonian eigenvalues corresponding to this violation.



(a) Asymptotic violation in (ϑ, ω) plane



(b) Hamiltonian eigenvalues corresponding to asymptotic violation for $\vartheta = \vartheta^*$

Figure 5.18: Asymptotic passivity characterization

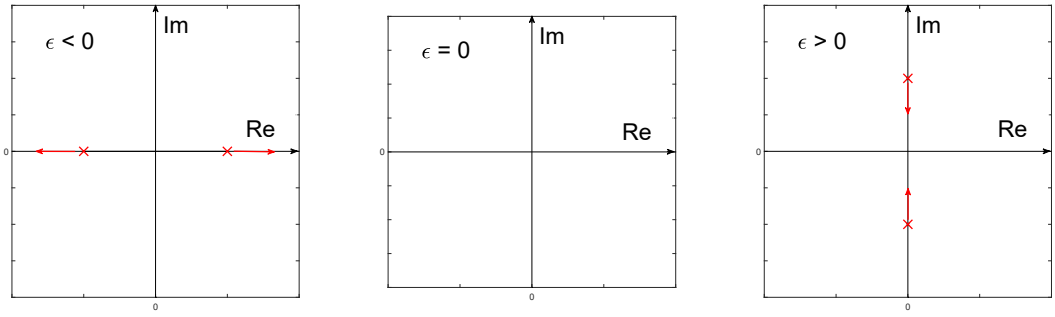
Denoting as $\vartheta = \vartheta^* + \epsilon$, with $\epsilon \in \mathbb{R}$, a generic parameter value, we can study what happens for different values of ϵ. Considering a pair of Hamiltonian eigenvalues we can identify three different conditions:

1. $\epsilon < 0$: The model is passive by assumption so no purely imaginary eigenvalues are present (see Figure 5.19a);

2. $\epsilon = 0$: The model is not passive for $\omega \to \infty$, which means that there are purely imaginary Hamiltonian eigenvalues with infinitely large magnitude (see Figure 5.19b);

3. $\epsilon > 0$: In this region the model is not passive thus there are finite purely imaginary Hamiltonian eigenvalues (see Figure 5.19c).

Recalling that the Hamiltonian eigen-spectrum has a 4-quadrant symmetry, the only way through which a real eigenvalue can become imaginary (with infinitely large magnitude), is to follow an unbounded trajectory that, ideally, connects together real and imaginary axes. This consideration leads us to conclude that, as the passive model approaches the asymptotic violation area, the eigenvalues real part increases toward infinity.
To better visualize what happens, a good insight is given by the eigenvalues projection onto the Riemann's sphere. In Figure 5.20a and 5.20b we show the behaviour of a Hamiltonian eigenvalue $\lambda_i(\vartheta)$ for increasing parameter values: we are interested to study the intersections of the green line, that connects the eigenvalue position on the complex plane and the Riemann's sphere top, with the sphere itself. This method is very useful in this scope because, by definition, it is possible to map the whole open complex plane on the



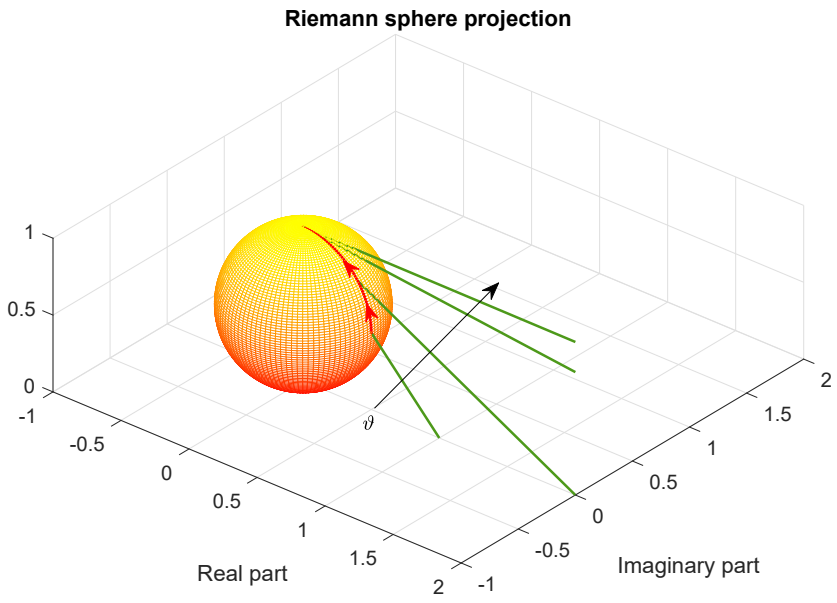
(a) Hamiltonian eigenvalues with negative epsilon (b) Hamiltonian eigenvalues with vanishing epsilon (c) Hamiltonian eigenvalues with positive epsilon

Figure 5.19: Riemann sphere eigenvalues projections

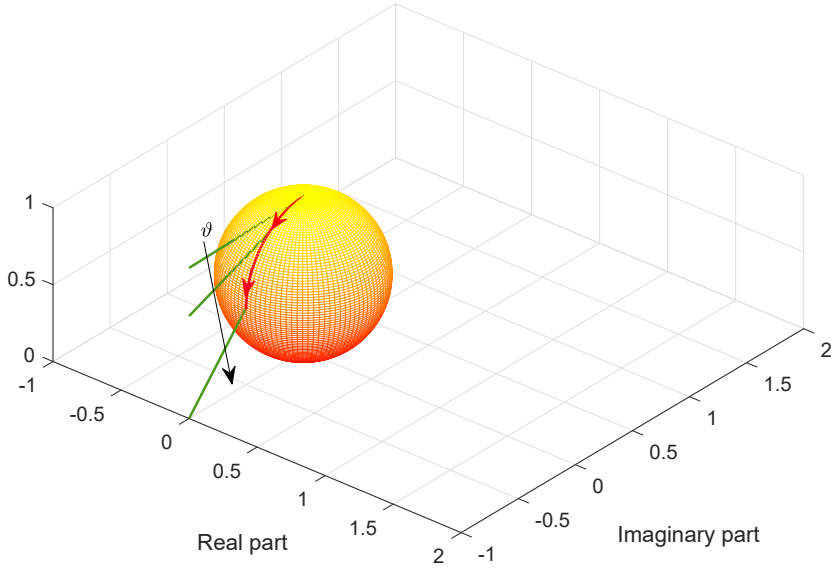
Riemann's sphere, enabling to deal with infinitely large quantities.
Initially $\lambda_i(\vartheta)$ is purely real and for increasing parameter its real part increases correspondingly, as the 5.20a shows. It is important to notice that, as the eigenvalue moves away from the origin, the green line becomes more and more horizontal, making the intersection point to converge toward the top of the sphere. Once $\mathrm{Re}\{\lambda_i(\vartheta)\} \to \infty$ for $\vartheta = \vartheta^*$, its projection on the sphere lays exactly on the north-pole, being the connection line perfectly collinear with the real axis. As the parameter value increases by an infinitesimal quantity, the projection continuously moves on the sphere and, due to spectrum symmetry, makes the green line to rotate by 90 degrees, leading to the phenomenon discussed above, as shown in Figure 5.20b
Thinking at the eigenvalue projection on the Riemann's sphere instead of at its position on the complex plane avoids the introduction of an unbounded trajectory that $\lambda_i(\vartheta)$ must follow.
As one can imagine by looking at the Riemann's sphere projections, as one eigenvalue $\lambda_i(\vartheta)$ is approaching the violation area, its velocity $v_{\lambda_i}(\vartheta)$, intended as:

$$v_{\lambda_i}(\vartheta) = \frac{\partial |\lambda_i(\vartheta)|}{\partial \vartheta}$$

blows up to infinity.
This consideration enables us to cast the search for asymptotic violations, in the derivative-based framework, as the search for parameters areas in which the velocity of at least one eigenvalue (and the one of its mirror image) increases by orders of magnitude.
Unfortunately this scheme is not suitable for our purposes because, even if not common, there are situations in which the velocity of some eigenvalues blows up without, however, being related to an asymptotic passivity violation. This would lead to sampling non-critical areas, slowing down the whole adaptive sampling process. For this reason, more refined techniques to detect asymptotic violations for parametric macro-models are required.
In the following we will go through two of these. The first one deals with the case in which the model denominator vanishes for some parameter value, while the other enables



(a) Riemann sphere projections for $\epsilon < 0$



(b) Riemann sphere projection for $\epsilon > 0$

Figure 5.20: Riemann's sphere projections

to find violations along the parameter axis, for a fixed frequency.

5.5.1 Vanishing Denominator

As anticipated, a vanishing denominator leads to an asymptotic passivity violation. In fact, denoting with ϑ^* the parameter value for which the denominator vanishes, the transfer function, when evaluated for ϑ^*, blows up to infinity. This means that, for $\vartheta = \vartheta^*$ at least one Hamiltonian eigenvalue (and its mirror image) is purely imaginary with infinite magnitude, as we discussed before. This fact can be seen on the singular values of the direct coupling matrix $\mathbf{D}(\vartheta)$ associated with the system realization, that for $\vartheta = \vartheta^*$ blow up, as Figure 5.21 shows:

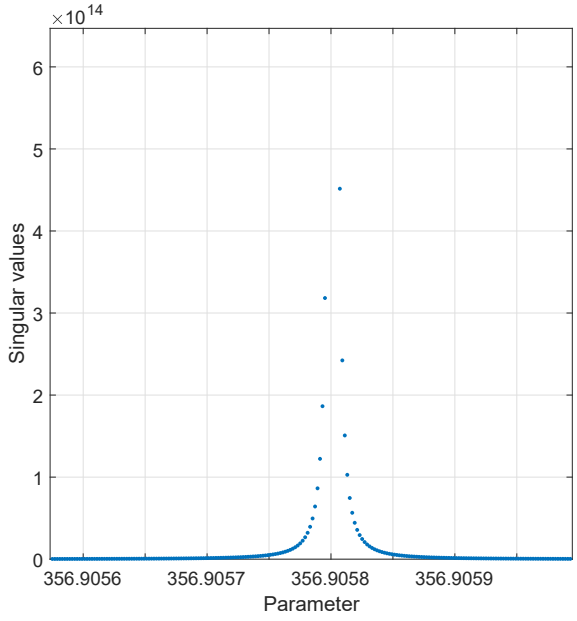


Figure 5.21: Singular values on singular model (*Test Case 14*)

Unfortunately, at this stage, nothing can be done to fix this issue. In fact we would need to act on denominator residues and perturb them to guarantee that no zeros are present in the parameter space, without losing accuracy. However, in the proposed passivity enforcement scheme we are allowed to modify just numerator residues (se Section 3.1), keeping the denominator ones fixed.
There exist advanced parametric model identification techniques that, to ensure uniform stability throughout the parameter space, enforce the denominator real part to be positive for any feasible parameter value (see [9], [8], [22]). A model identified with these methods, for sure, will not present this kind of issues.

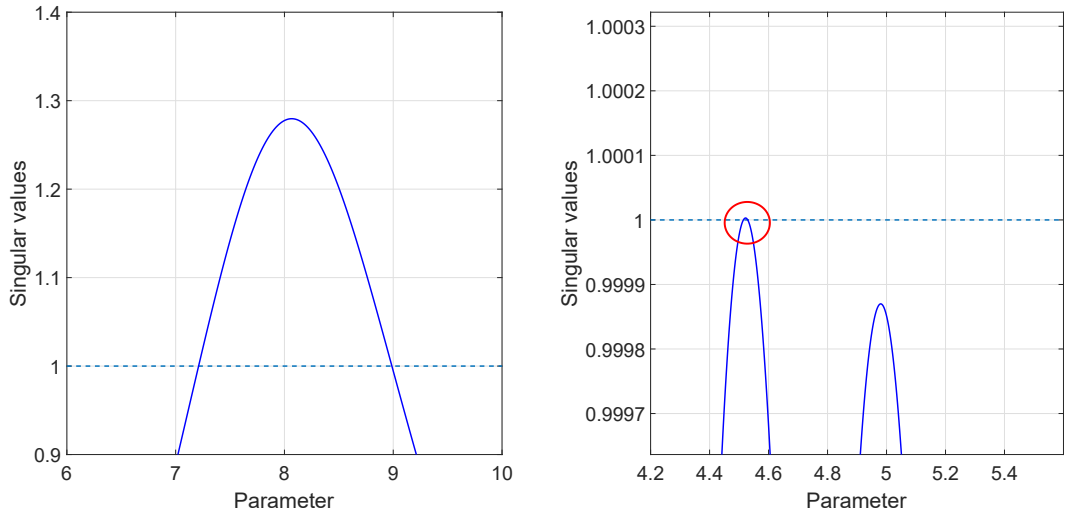
5.5.2 Model Passivity Violation

If the model denominator does not have singularities, we have an asymptotic passivity violation when $\mathbb{I} - \mathbf{D}(\vartheta)^\mathsf{T}\mathbf{D}(\vartheta)$ or $\mathbf{D}(\vartheta) + \mathbf{D}(\vartheta)^\mathsf{T}$ are not positive semi-definite for some parameter value, for scattering and immittance systems respectively. In the following

paragraphs we derive an algebraic approach to detect asymptotic violations for immittance and scattering representations.

Scattering Systems

Looking at $\mathbf{D}(\vartheta)$ matrix singular values, two situations may occur:



(a) Non-critical large passivity violation

(b) Critical small passivity violation

Figure 5.22: Asymptotic violations comparison

The case shown in 5.22a it is not critical, since any parametric passivity verification algorithm is able to detect the variation on the number of Hamiltonian imaginary eigenvalues, thus sampling accordingly the parameter space. Figure 5.22b shows instead a more critical case because the passivity violation is located in a small parameter space area, highlighted by the red circle. Being these asymptotic violations detectable just by the difference between the number of Hamiltonian imaginary eigenvalues at different sampling points, a standard passivity check algorithm may not spot them. For this reason, some violations similar to the latter may not be detected and more refined techniques must be used.
In the following, we are going to present an algebraic method that is able to detect for which parameter a singular value of $\mathbf{D}(\vartheta)$ intersects the unity threshold, detecting the opening or closing of a violation area.
To this end, recalling that $\mathbf{D}(\vartheta)$ is real-valued, we focus on the following equation:

$$\mathbb{I} - \mathbf{D}(\vartheta)^{\mathsf{T}}\mathbf{D}(\vartheta) = 0$$

Recalling the definition of singular values, the previous relation can be cast (for unitary singular values) as:

$$\begin{cases} \mathbf{D}(\vartheta) \cdot \boldsymbol{u} = \boldsymbol{v} \\ \mathbf{D}^{\mathsf{T}}(\vartheta) \cdot \boldsymbol{v} = \boldsymbol{u} \end{cases}$$

and, using the expression of $\mathbf{D}(\vartheta)$ in terms of model residues (see Section 2.1.2), we get:

$$\begin{cases} \frac{\mathbf{R}_0(\vartheta)}{r_0(\vartheta)} \cdot \boldsymbol{u} = \boldsymbol{v} \\ \\ \frac{\mathbf{R}_0^{\mathsf{T}}(\vartheta)}{r_0(\vartheta)} \cdot \boldsymbol{v} = \boldsymbol{u} \end{cases} \tag{5.25}$$

where $\mathbf{R}_0(\vartheta)$ and $r_0(\vartheta)$ are, respectively, the numerator and denominator residues associated with the partial fraction basis $\varphi_0(s) = 1$. Recalling that $\bar{\ell}_N$ and $\bar{\ell}_D$ are the parameter basis orders for numerator and denominator respectively, it holds that:

$$\begin{cases} \mathbf{R}_0(\vartheta) = \sum_{\ell=0}^{\bar{\ell}_N} \mathbf{R}_{0,\ell} \xi_\ell(\vartheta) \\ \\ r_0(\vartheta) = \sum_{\ell=0}^{\bar{\ell}_D} r_{0,\ell} \xi_\ell(\vartheta) \end{cases}$$

it is then possible to expand Equation (5.25) as:

$$\begin{cases} \sum_{\ell=0}^{\bar{\ell}_N} \mathbf{R}_{0,\ell} \cdot \boldsymbol{u}_\ell - \sum_{\ell=0}^{\bar{\ell}_D} r_{0,\ell} \cdot \boldsymbol{v}_\ell = 0 \\ \\ \sum_{\ell=0}^{\bar{\ell}_N} \mathbf{R}_{0,\ell}^{\mathsf{T}} \cdot \boldsymbol{v}_\ell - \sum_{\ell=0}^{\bar{\ell}_D} r_{0,\ell} \cdot \boldsymbol{u}_\ell = 0 \end{cases} \tag{5.26}$$

with $\boldsymbol{v}_\ell(\vartheta) = \xi_\ell(\vartheta)\boldsymbol{v}$ and $\boldsymbol{u}_\ell(\vartheta) = \xi_\ell(\vartheta)\boldsymbol{u}$. We omit the dependency on ϑ for readability. Up to now the basis functions $\xi_\ell(\vartheta)$ are generic. In the following we will focus on orthogonal polynomials basis functions. We know from section 2.1.1 that a generic orthogonal polynomial can be generated through a recurrence relation, with suitable weights α_ℓ, β_ℓ and δ_ℓ, that reads:

$$\xi_{\ell+1}(\vartheta) = (\alpha_\ell \vartheta + \beta_\ell)\xi_\ell(\vartheta) + \delta_{\ell-1}\xi_{\ell-1}(\vartheta)$$

In this case, vectors $\boldsymbol{u}_\ell$ and $\boldsymbol{v}_\ell$ can be recursively written, up to order $\bar{\ell} - 1$ as:

$$\begin{cases} \boldsymbol{u}_1 = \alpha_0 \vartheta \boldsymbol{u}_0 + \beta_0 \boldsymbol{u}_0 \\ \boldsymbol{u}_2 = (\alpha_1 \vartheta + \beta_1)\boldsymbol{u}_1 + \delta_0 \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{\bar{\ell}-1} = (\alpha_{\bar{\ell}-2} \vartheta + \beta_{\bar{\ell}-2})\boldsymbol{u}_{\bar{\ell}-2} + \delta_{\bar{\ell}-3}\boldsymbol{u}_{\bar{\ell}-3} \\ \boldsymbol{v}_1 = \alpha_0 \vartheta \boldsymbol{v}_0 + \beta_0 \boldsymbol{v}_0 \\ \boldsymbol{v}_2 = (\alpha_1 \vartheta + \beta_1)\boldsymbol{v}_1 + \delta_0 \boldsymbol{v}_0 \\ \vdots \\ \boldsymbol{v}_{\bar{\ell}-1} = (\alpha_{\bar{\ell}-2} \vartheta + \beta_{\bar{\ell}-2})\boldsymbol{v}_{\bar{\ell}-2} + \delta_{\bar{\ell}-3}\boldsymbol{v}_{\bar{\ell}-3} \end{cases} \tag{5.27}$$

so that the terms $\boldsymbol{u}_{\bar{\ell}}$ and $\boldsymbol{v}_{\bar{\ell}}$, through the previous set of equations, depend linearly on the parameter ϑ, as shown below:

$$\boldsymbol{u}_{\bar{\ell}} = (\alpha_{\bar{\ell}-1} \vartheta + \beta_{\bar{\ell}-1})\boldsymbol{u}_{\bar{\ell}-1} + \delta_{\bar{\ell}-2}\boldsymbol{u}_{\bar{\ell}-2}$$
$$\boldsymbol{v}_{\bar{\ell}} = (\alpha_{\bar{\ell}-1} \vartheta + \beta_{\bar{\ell}-1})\boldsymbol{v}_{\bar{\ell}-1} + \delta_{\bar{\ell}-2}\boldsymbol{v}_{\bar{\ell}-2}$$

Thus Equation (5.26) can be re-written as

$$\begin{cases} \sum_{\ell=0}^{\bar{\ell}_N-1} \mathbf{R}_{0,\ell} \cdot \boldsymbol{u}_\ell + \mathbf{R}_{0,\bar{\ell}_N} \cdot \boldsymbol{u}_{\bar{\ell}_N} - \sum_{\ell=0}^{\bar{\ell}_D-1} r_{0,\ell} \cdot \boldsymbol{v}_\ell - r_{0,\bar{\ell}_D} \cdot \boldsymbol{v}_{\bar{\ell}_D} = 0 \\ \sum_{\ell=0}^{\bar{\ell}_N-1} \mathbf{R}_{0,\ell}^{\mathsf{T}} \cdot \boldsymbol{v}_\ell + \mathbf{R}_{0,\bar{\ell}_N}^{\mathsf{T}} \cdot \boldsymbol{v}_{\bar{\ell}_N} - \sum_{\ell=0}^{\bar{\ell}_D-1} r_{0,\ell} \cdot \boldsymbol{u}_\ell - r_{0,\bar{\ell}_D} \cdot \boldsymbol{u}_{\bar{\ell}_D} = 0 \end{cases} \tag{5.28}$$

where $\boldsymbol{u}_{\bar{\ell}_N}$, $\boldsymbol{u}_{\bar{\ell}_D}$, $\boldsymbol{v}_{\bar{\ell}_N}$ and $\boldsymbol{v}_{\bar{\ell}_D}$ are explicit and depends linearly on the parameter ϑ through (5.27).
The set of equations formed by (5.28) and (5.27) is thus affine in the parameter. By re-arranging properly the terms, it is possible to solve this system for ϑ. The re-arranged equations are shown in Equation (5.29):

$$\begin{cases} [\mathbb{W}_R^{\bar{\ell}_N},\ -\mathbb{W}_{\mathbb{I}_P r}^{\bar{\ell}_D}] \cdot \mathbf{z} + \vartheta \cdot [\mathbb{S}_R^{\bar{\ell}_N},\ -\mathbb{S}_{\mathbb{I}_P r}^{\bar{\ell}_D}] \cdot \mathbf{z} = 0 \\ [-\mathbb{W}_{\mathbb{I}_P r}^{\bar{\ell}_D},\ \mathbb{W}_{R^{\mathsf{T}}}^{\bar{\ell}_N}] \cdot \mathbf{z} + \vartheta \cdot [-\mathbb{S}_{\mathbb{I}_P r}^{\bar{\ell}_D},\ \mathbb{S}_R^{\bar{\ell}_N}] \cdot \mathbf{z} = 0 \end{cases} \tag{5.29}$$

with

$$\mathbb{W}_R^{\bar{\ell}_N} = [\mathbf{R}_{0,0},\ \mathbf{R}_{0,1}, \ldots, \mathbf{R}_{0,\bar{\ell}_N-2} + \mathbf{R}_{0,\bar{\ell}_N}\delta_{\bar{\ell}_N-1},\ \ \mathbf{R}_{0,\bar{\ell}_N-1} + \mathbf{R}_{0,\bar{\ell}_N}\beta_{\bar{\ell}_N-1},\ \mathbb{Z}_N]$$

$$\mathbb{W}_{\mathbb{I}_P r}^{\bar{\ell}_D} = [\mathbb{I}_{\mathbf{P}} r_{0,0},\ \mathbb{I}_{\mathbf{P}} r_{0,1}, \ldots, \mathbb{I}_{\mathbf{P}} r_{0,\bar{\ell}_D-2} + \mathbb{I}_{\mathbf{P}} r_{0,\bar{\ell}_D}\delta_{\bar{\ell}_D-1},\ \ \mathbb{I}_{\mathbf{P}} r_{0,\bar{\ell}_D-1} + \mathbb{I}_{\mathbf{P}} r_{0,\bar{\ell}_D}\beta_{\bar{\ell}_D-1},\ \mathbb{Z}_D]$$

$$\mathbb{S}_R^{\bar{\ell}_N} = [\mathbf{0}, \mathbf{0}, \ldots, \alpha_{\bar{\ell}_N-1}\mathbf{R}_{\bar{\ell}_N},\ \mathbb{Z}_N]$$

$$\mathbb{S}_{\mathbb{I}_P r}^{\bar{\ell}_D} = [\mathbf{0}, \mathbf{0}, \ldots, \alpha_{\bar{\ell}_D-1}\mathbf{R}_{\bar{\ell}_D},\ \mathbb{Z}_D]$$

$$\mathbf{z} = [\boldsymbol{u}_0^{\mathsf{T}}, \ldots, \boldsymbol{u}_{\bar{\ell}-1}^{\mathsf{T}}, \boldsymbol{v}_0^{\mathsf{T}}, \ldots, \boldsymbol{v}_{\bar{\ell}-1}^{\mathsf{T}}]^{\mathsf{T}},\ \bar{\ell} = \max\{\bar{\ell}_N,\ \bar{\ell}_D\}$$

where matrices $\mathbb{Z}_N$ and $\mathbb{Z}_D$ take into account the case in which $\bar{\ell}_N \neq \bar{\ell}_D$.
More precisely, if $\bar{\ell}_N > \bar{\ell}_D$:

$$\mathbb{Z}_D = \mathbf{0} \in \mathbb{R}^{P \times P(\bar{\ell}_N - \bar{\ell}_D)}$$
$$\mathbb{Z}_N \text{ is empty}$$

conversely, if $\bar{\ell}_D > \bar{\ell}_N$:

$$\mathbb{Z}_N = \mathbf{0} \in \mathbb{R}^{P \times P(\bar{\ell}_D - \bar{\ell}_N)}$$
$$\mathbb{Z}_D \text{ is empty}$$

Finally, if $\bar{\ell}_N = \bar{\ell}_D$, both $\mathbb{Z}_D$ and $\mathbb{Z}_N$ will be empty.

Defining now the following matrices:

$$\boldsymbol{\Xi}_{\bar{\ell}} = \begin{pmatrix} \beta_0\mathbb{I}_P & -\mathbb{I}_P & \mathbf{0}^P & & \\ \delta_1\mathbb{I}_P & \beta_1\mathbb{I}_P & -\mathbb{I}_P & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{\bar{\ell}-2}\mathbb{I}_P & \beta_{\bar{\ell}-2}\mathbb{I}_P & -\mathbb{I}_P \end{pmatrix}$$

$$\boldsymbol{\Psi}_{\bar{\ell}} = \begin{pmatrix} \alpha_0\mathbb{I} & & & \mathbf{0}^P \\ & \alpha_1\mathbb{I} & & \mathbf{0}^P \\ & & \ddots & \vdots \\ & & \alpha_{\bar{\ell}-2} & \mathbf{0}^P \end{pmatrix}$$

the complete set of equation is:

$$\begin{cases} \begin{pmatrix} \mathbb{W}_R^{\bar{\ell}_N} & -\mathbb{W}_{\mathbb{I}_P r}^{\bar{\ell}_D} \\ \Xi_{\bar{\ell}} & \mathbf{0}^{P(\bar{\ell}-2)} \end{pmatrix} \cdot \mathbf{z} + \vartheta \cdot \begin{pmatrix} \mathbb{S}_R^{\bar{\ell}_N} & -\mathbb{S}_{\mathbb{I}_P r}^{\bar{\ell}_D} \\ \Psi_{\bar{\ell}} & \mathbf{0}^{P(\bar{\ell}-2)} \end{pmatrix} \cdot \mathbf{z} = \mathbf{0}^{P\bar{\ell}\times 1} \\ \begin{pmatrix} -\mathbb{W}_{\mathbb{I}_P r}^{\bar{\ell}_D} & \mathbb{W}_{R^\intercal}^{\bar{\ell}_N} \\ \mathbf{0}^{P(\bar{\ell}-2)} & \Xi_{\bar{\ell}} \end{pmatrix} \cdot \mathbf{z} + \vartheta \cdot \begin{pmatrix} -\mathbb{S}_{\mathbb{I}_P r}^{\bar{\ell}_D} & \mathbb{S}_{R^\intercal}^{\bar{\ell}_N} \\ \mathbf{0}^{P(\bar{\ell}-2)} & \Psi_{\bar{\ell}} \end{pmatrix} \cdot \mathbf{z} = \mathbf{0}^{P\bar{\ell}\times 1} \end{cases}$$

Looking at its particular structure, the above system can be cast in just one block equation as follows:

$$\mathbb{P}_0 \cdot \mathbf{z} + \vartheta \mathbb{P}_1 \cdot \mathbf{z} = \mathbf{0}^{2P\bar{\ell}\times 1} \tag{5.30}$$

where the matrices $\mathbb{P}_0$ and $\mathbb{P}_1$ are:

$$\mathbb{P}_0 = \begin{pmatrix} \mathbb{W}_R^{\bar{\ell}_N} & -\mathbb{W}_{\mathbb{I}_P r}^{\bar{\ell}_D} \\ \Xi_{\bar{\ell}} & \mathbf{0}^{P(\bar{\ell}-2)} \\ -\mathbb{W}_{\mathbb{I}_P r}^{\bar{\ell}_D} & \mathbb{W}_{R^\intercal}^{\bar{\ell}_N} \\ \mathbf{0}^{P(\bar{\ell}-2)} & \Xi_{\bar{\ell}} \end{pmatrix} \in \mathbb{R}^{2P\bar{\ell}} \tag{5.31}$$

$$\mathbb{P}_1 = \begin{pmatrix} \mathbb{S}_R^{\bar{\ell}_N} & -\mathbb{S}_{\mathbb{I}_P r}^{\bar{\ell}_D} \\ \Psi_{\bar{\ell}} & \mathbf{0}^{P(\bar{\ell}-2)} \\ -\mathbb{S}_{\mathbb{I}_P r}^{\bar{\ell}_D} & \mathbb{S}_{R^\intercal}^{\bar{\ell}_N} \\ \mathbf{0}^{P(\bar{\ell}-2)} & \Psi_{\bar{\ell}} \end{pmatrix} \in \mathbb{R}^{2P\bar{\ell}}$$

Immittance Systems

An immittance system is not asymptotically passive if $\mathbf{D}(\vartheta) + \mathbf{D}^{\mathsf{T}}(\vartheta)$ is not positive-definite. In order to find asymptotic violations we need to find for which parameter values $\mathbf{D}(\vartheta) + \mathbf{D}^{\mathsf{T}}(\vartheta)$ is singular. To this end we can write:

$$\left[\mathbf{D}(\vartheta) + \mathbf{D}^{\mathsf{T}}(\vartheta)\right] \boldsymbol{u} = 0$$

for any vector $\boldsymbol{u}$.
We recall that:

$$\mathbf{D}(\vartheta) = \frac{\mathbf{R}_0(\vartheta)}{r_0(\vartheta)}$$

Assuming that $r_0(\vartheta) \neq 0 \; \forall \vartheta$, as it is always when dealing with guaranteed stable models [8, 9, 22], it follows that:

$$\left[\frac{\mathbf{R}_0(\vartheta)}{r_0(\vartheta)} + \frac{\mathbf{R}_0^{\mathsf{T}}(\vartheta)}{r_0(\vartheta)}\right] \boldsymbol{u} = 0 \;\Rightarrow \left[\mathbf{R}_0(\vartheta) + \mathbf{R}_0^{\mathsf{T}}(\vartheta)\right] \boldsymbol{u} = 0 \tag{5.32}$$

Defining $\mathbf{R}^{'}(\vartheta) = \mathbf{R}_0(\vartheta) + \mathbf{R}_0^{\mathsf{T}}(\vartheta)$, we know from Section 2.1 that:

$$\mathbf{R}^{'}(\vartheta) = \sum_{\ell=0}^{\bar{\ell}} (\mathbf{R}_{0,\ell}\xi_\ell(\vartheta) + \mathbf{R}_{0,\ell}^{\mathsf{T}}\xi_\ell(\vartheta)) = \sum_{\ell=0}^{\bar{\ell}} \mathbf{R}_{0,\ell}^{'}\xi_\ell(\vartheta)$$

where $\mathbf{R}_{0,\ell}^{'} = \mathbf{R}_{0,\ell} + \mathbf{R}_{0,\ell}^{\mathsf{T}}$ From (5.32) we can state that:

$$\left[\sum_{\ell=0}^{\bar{\ell}} \mathbf{R}_{0,\ell}^{'}\xi_\ell(\vartheta)\right] \boldsymbol{u} = 0 \Rightarrow \left[\sum_{\ell=0}^{\bar{\ell}} \mathbf{R}_{0,\ell}^{'}\boldsymbol{u}_\ell(\vartheta)\right] = 0 \tag{5.33}$$

with $\boldsymbol{u}_\ell(\vartheta) = \boldsymbol{u}\ \xi_\ell(\vartheta)$.
As in the scattering case, using the orthogonal polynomials recursive relation we can write vectors $\boldsymbol{u}_\ell$ up to the order $\bar{\ell} - 1$ as:

$$\begin{cases} \boldsymbol{u}_1 = \alpha_0\vartheta\boldsymbol{u}_0 + \beta_0\boldsymbol{u}_0 \\ \boldsymbol{u}_2 = (\alpha_1\vartheta + \beta_1)\boldsymbol{u}_1 + \delta_0\boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{\bar{\ell}-1} = (\alpha_{\bar{\ell}-2}\vartheta + \beta_{\bar{\ell}-2})\boldsymbol{u}_{\bar{\ell}-2} + \delta_{\bar{\ell}-3}\boldsymbol{u}_{\bar{\ell}-3} \end{cases} \tag{5.34}$$

Thus, through the previous relations, the term $\boldsymbol{u}_{\bar{\ell}}$ depends linearly on the parameter ϑ. We can expand (5.33) as:

$$\sum_{\ell=0}^{\bar{\ell}_N - 1} \mathbf{R}_{0,\ell}^{'} \cdot \boldsymbol{u}_\ell + \mathbf{R}_{0,\bar{\ell}_N}^{'} \cdot \boldsymbol{u}_{\bar{\ell}_N} = 0 \tag{5.35}$$

The last term is linearly dependent on ϑ through (5.34). The set of Equations defined by (5.35) and (5.34) is now affine in the parameter ϑ. Thanks to this consideration it is

possible to cast the problem of finding asymptotic violations as we did in the scattering case. Moreover, in this case, the matrices that are involved are much simpler.
By re-arranging the terms in (5.35) and (5.34), we get:

$$\mathbb{P}_0\mathbf{z} + \vartheta\ \mathbb{P}_1\mathbf{z} = 0 \tag{5.36}$$

where $\mathbf{z} = [\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{\bar{\ell}-1}]$, and

$$\mathbb{P}_0 = \begin{pmatrix} \mathbb{W}_{R'}^{\bar{\ell}_N} \\ \\ \Xi_{\bar{\ell}} \end{pmatrix} \qquad\qquad \mathbb{P}_1 = \begin{pmatrix} \mathbb{S}_{R'}^{\bar{\ell}_N} \\ \\ \Psi_{\bar{\ell}} \end{pmatrix}$$

Equations (5.30) and (5.36) form generalized eigenvalue problems, where the generalized eigenvalues are the parameter values for which at least a singular value of $\mathbf{D}(\vartheta)$ matrix intersects the unity threshold or an eigenvalue of $\mathbf{D}(\vartheta) + \mathbf{D}^{\mathsf{T}}(\vartheta)$ vanishes. The solution of these problems can be found with any available eigen-solver. In general, not all the generalized eigenvalues correspond to a passivity violation because they may be in complex-conjugate pairs (that are not feasible since the parameters of interest are real numbers) or be located outside the considered parameter space. Therefore we need to find the generalized eigenvalues $\vartheta_i \in \Theta$.
In the following we present a set of numerical results that test this algorithm on a number of real test cases. We report results for scattering cases, but the same holds for immittance systems. In the figures, blue lines represent the singular values of $\mathbf{D}(\vartheta)$ computed for a dense parameter sweep, while yellow dots correspond to unit threshold intersection computed with the proposed method.
In Figure 5.23 we propose a result taken from *Test Case 2–b* (see Appendix A), where the asymptotic passivity violation spans a large amount of the parameter space, then it is well detectable by the standard passivity check algorithm and no refined algebraic checks are required. However, we see that this method is able to spot precisely where asymptotic violations occur.

Figure 5.24 shows, instead, a more critical case (taken from *Test Case 4–b*) where, as explained above, an algebraic test is required. In fact the violation is located in a small parameter space region (the parameter space is defined from $\vartheta = 1$ to $\vartheta = 25$, then the violation spans just a 0.06 % of the whole space), that may not be detected by the standard passivity verification algorithm. We see that the proposed scheme is able to detect violation points.

Figures 5.25a shows an even more interesting test case (*Test Case 4–b*), in which it seems that the proposed method fails. In fact, there is a solution of (5.30) corresponding to $\vartheta \approx 24.8$ without any singular value crossing the unit threshold. Actually, by zooming in this area, as shown in Figure 5.25b, we see that the algebraic approach returns a correct result and the above problem is due to poor number of sampled singular values.

If violations are found, we need to formulate additional passivity constraints. To this end, as for the classical Hamiltonian driven method performed along the frequency axis,

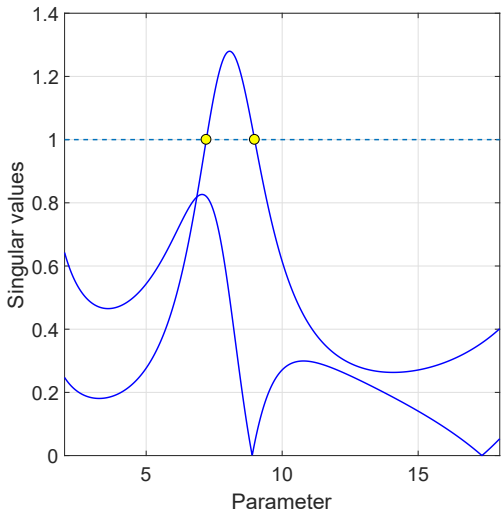


Figure 5.23: Well behaved singular values (*Test Case 2–b*)

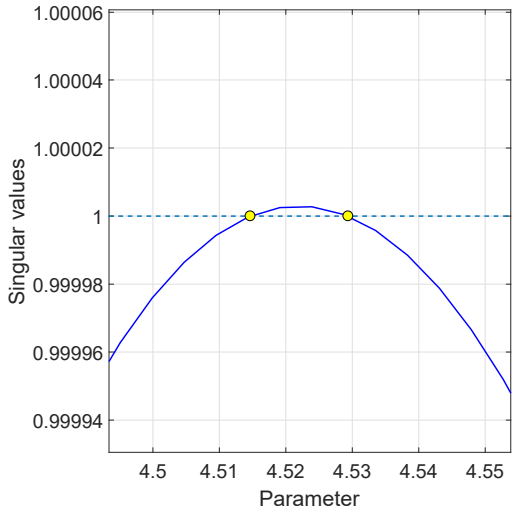
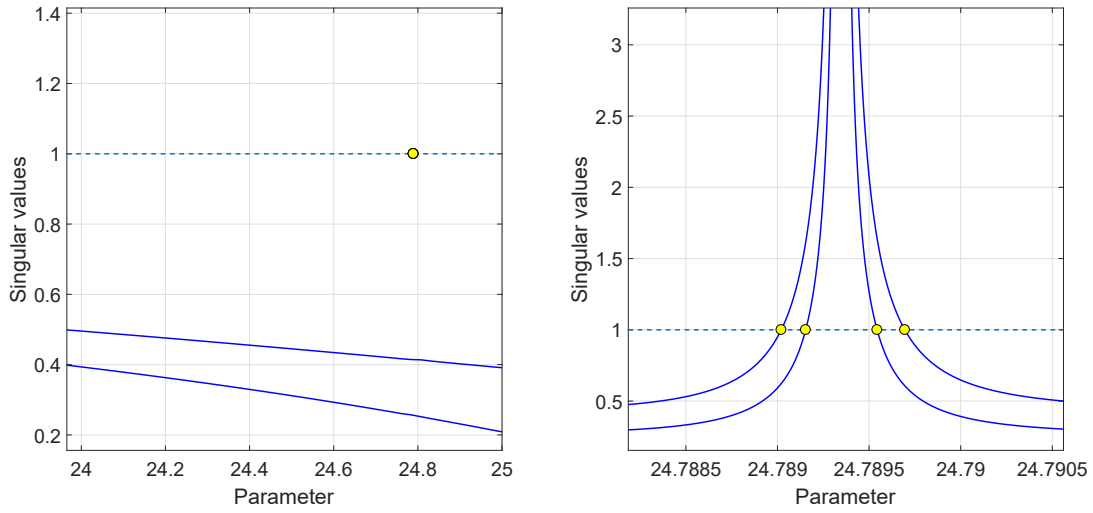


Figure 5.24: Critical singular values (*Test Case 4–b*)

we partition the parameter space accordingly to the solutions of (5.30) or (5.36) and, by extracting the singular values of $\mathbf{D}(\vartheta)$ or the eigenvalues of $\mathbf{D}(\vartheta) + \mathbf{D}(\vartheta)^{\mathsf{T}}$ in the middle of each violation sub-interval we can detect which is passive and which is not. In the non-passive ones, a set of additional points are sampled and the constraints are formulated. Algorithm 5.10 shows this procedure.



(a) Seemingly wrong solution

(b) Zoom on the same area

Figure 5.25: An example of the algebraic method accuracy (*Test Case 4–b*)

Algorithm 5.10 Asymptotic passivity violations detection

1: Get model numerator and denominator residues $\mathbf{R}_{n,\ell}$, $r_{n,\ell}$ for $n = 0, \ldots, \bar{n}$, $\ell = 0, \ldots, \bar{\ell}$
2: Build matrices $\mathbb{P}_0$ and $\mathbb{P}_1$ according to (5.31)
3: Solve the eigenproblem in (5.30) or (5.36) for $\vartheta \in \Theta$ and collect violations in a vector τ
4: **if** $\mathbf{card}(\tau) = 0$ **then**
5: **break**
6: **else**
7: Partition the parameter axis in $\mathbf{card}(\tau) + 1$ sub-intervals according to the violations found before
8: Find n_v regions where the model is not asymptotically passive through $\mathbf{D}(\vartheta)$ singular values or $\mathbf{D}(\vartheta)^{\mathsf{T}} + \mathbf{D}(\vartheta)$ eigenvalues
9: **for** i $= 1 : n_v$ **do**
10: Get left and right edges $[\vartheta_{L_i}, \vartheta_{R_i}]$ of the i-th violation interval
11: Sample this region with additional points as detailed in Section 5.2
12: Update violation structure $\mathcal{W}$
13: **end for**
14: **end if**

Chapter 6

Multi-Variate Passivity Verification

In previous Chapters the problems of passivity verification and enforcement for parametric macro-models has been addressed. However we restricted our framework to non-parametric (just frequency dependent) or mono-parametric models (dependent on the frequency and an external parameter). In this Chapter we are going to extend our framework to multi-parametric models. In particular, we will discuss about some issues that come along with the increasing number of parameters and we will present a passivity verification method for bi-variate macro-models (dependent on the frequency and two external parameters). Finally, several numerical results show the capabilities of the proposed scheme.

6.1 Multi-Parametric Passivity Assessment

In Chapters 4 and 5 we saw that, as the number of parameters increases, the complexity in developing passivity verification algorithms increases correspondingly due to the necessity of finding a suitable trade-off between reliability and computational times. In fact, since no algebraic approaches are available to detect passivity violations in a $(\rho+1)$-dimensional space, with ρ the number of parameters, all we can do is to adaptively move along the parameters directions while performing Hamiltonian-driven tests along the frequency axis. As one can imagine, as the number of parameters increase, this task becomes more and more complex, up to be un-manageable in terms of memory and computational time requirements. In technical literature this fact is known with the name of "*curse of dimensionality*". In details, the curse of dimensionality arises whenever we deal with high dimensional problems and, in our case, is related to the number of samples required to reliably check the model passivity on a multi-dimensional space, that grows exponentially with the number of parameters. In fact, supposing that each parameter is orthogonal with respect to the others, the parameter space is a hypercube in $\mathbb{R}^\rho$. Thus, assuming to place a number f of samples along each dimension, we see that the total number of points is $F = f^\rho$. Recalling that for each sample we must extract Hamiltonian eigenvalues, it follows that computational times blow up even with low values of ρ. These considerations

lead us to conclude that the problem of multi-parametric passivity verification is, in general, too hard to be solved with adaptive techniques and that more refined schemes are needed. On the other hand, with $\rho = 2$ the computational effort is expected to be still reasonable. Thus, in the following we will develop an adaptive passivity verification algorithm for bi-variate macro-models.

6.2 A Bi-Dimensional Parameter Space

In the case of a bi-variate model, the space in which we must verify passivity is a subset of $\mathbb{R}^3$. Assuming to use a Hamiltonian-verification method along the frequency axis, we need to adaptively search for violations no more on a segment, as in the mono-dimensional case, but on a bounded subset Θ of $\mathbb{R}^2$, see (2.1). Some observations about this plane are in order. From now on, in fact, we will assume that the parameters are independent, so that the variation of one parameter does not affect the other. This is compliant with the parameters space definition given in (2.1), that leads Θ to be at most rectangular. Additionally, with the idea of using a hierarchical approach to define the points in Θ, as detailed in Section 5.3.1, we will normalize its boundaries to be $[0,\ 1]$ segments, reducing it to a unit area square, denoted as $\widetilde{\Theta}$. In the following, we will label the two parameters, without loss of generality, with x and y. Thus, a point $\tilde{\boldsymbol{\vartheta}}_{\mathbf{m}} = [\tilde{\vartheta}^{(x)}_{m_x},\ \tilde{\vartheta}^{(y)}_{m_y}]^{\mathsf{T}}$ belonging to the normalized space $\widetilde{\Theta}$ is uniquely determined by the value attained by the multi-index $\mathbf{m} = [m_x,\ m_y]^{\mathsf{T}}$. Coordinates $\tilde{\vartheta}^{(x)}_{m_x}$ and $\tilde{\vartheta}^{(y)}_{m_y}$ are identified in the hierarchical framework by indices $\{k_{m_x},\ j_{m_x}\}$ and $\{k_{m_y},\ j_{m_y}\}$ as:

$$\begin{aligned} \tilde{\vartheta}^{(x)}_{m_x} &= k_{m_x} \cdot 2^{-j_{m_x}} \\ \tilde{\vartheta}^{(y)}_{m_y} &= k_{m_y} \cdot 2^{-j_{m_y}} \end{aligned} \tag{6.1}$$

As for the mono-parametric case, we partition the normalized parameter space $\widetilde{\Theta}$ in smaller areas, from now on denoted as *"patches"*, used as elementary objects where, with a suitable strategy, we will search for passivity violations (in the following we will refer to this practice also as "*tessellation*"). About the shape of these patches, we must be aware that the refinement algorithm performances strongly depend on it. We know from finite-elements literature that is common practice to use triangle meshes due to some peculiar properties that they have but, in our case, being the domain $\widetilde{\Theta}$ a square, we can safely use squares or, more in general, rectangles. This choice, as we will see later on, greatly simplifies the algorithm implementation without reducing its reliability.
A patch is completely defined by the coordinates of its four vertices: in fact, with just these information we can identify uniquely every patch in the parameter space. Starting from the left-bottom vertex and moving in counter-clockwise direction, the vertices are denoted as $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$, $\mathcal{V}_4$. A graphical representation is given in Figure 6.1

If a set of non-passive parameter values is likely to be found inside a patch, the latter must be refined to better characterize the area. For us, refining a patch means to subdivide it in smaller patches that, inheriting all the properties detailed above, must be at most rectangles.

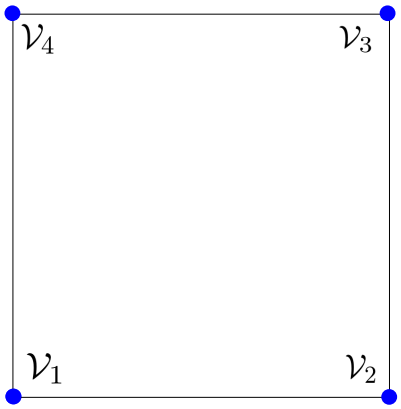
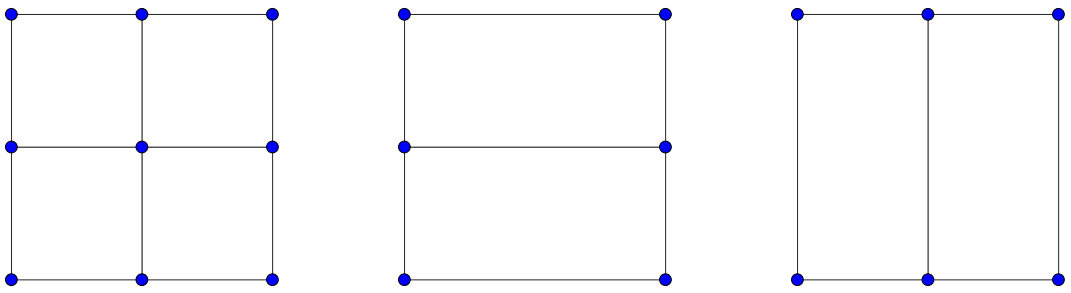


Figure 6.1: Elementary patch with vertices location

If, in addition to that, we notice from (6.1) that a bisection approach is used to define points along both axes, just three different types of patch subdivision are allowed:

- Four sub-squares (see Figure 6.2a). When this refinement is performed all the edges mid-points and the patch center are sampled. In the following we will denote this refinement as "*full*";
- Two rectangles whose length is twice the height (see Figure 6.2b). This refinement requires to sample just the vertical edges mid-points. In the following this refinement will be denoted as "*vertical*";
- Two rectangles whose height is twice the length (see Figure 6.2c). This refinement requires to sample just the horizontal edges mid-points. In the following this refinement will be denoted as "*horizontal*".



(a) Full refinement (b) Vertical refinement (c) Horizontal refinement

Figure 6.2: Patch refinement schemes

To better explain the refinement procedure we want to pursue, Figure 6.3 shows an example of two successive refinements, identified by different colors. We start with a 2×2 uniform grid, whose vertices are identified by blue dots, then we proceed by refining

horizontally the left-bottom patch, sampling the red-dotted points. Then, the resulting rectangular patch is then fully refined, which means that all the five green-dotted points are sampled.

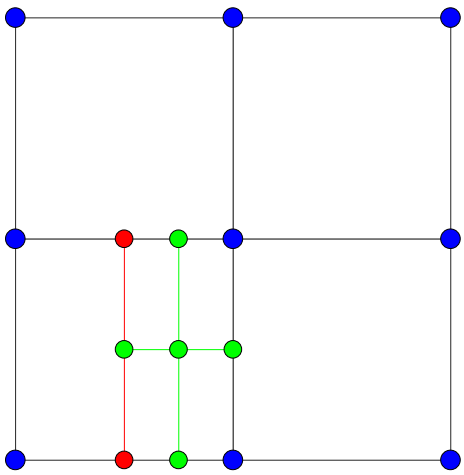


Figure 6.3: Successive refinements on the normalized parameter plane $\widetilde{\Theta}$

We assume now that the vertices locations for a generic patch on the parameters space are known and defined by a multi-index $\mathbf{m}_i = [m_{i_x},\ m_{i_y}]^\mathsf{T}$, such that the i-th vertex $\mathcal{V}_i$ is defined as:

$$\mathcal{V}_i = \tilde{\boldsymbol{\vartheta}}_{\mathbf{m}_i} = \{\tilde{\vartheta}^{(x)}_{m_{i_x}},\ \tilde{\vartheta}^{(y)}_{m_{i_y}}\}$$

for $i = 1, \ldots, 4$.
For readability, from now on, we will drop the subscript m and superscripts (x), (y), thus:

$$\mathcal{V}_i = \{\tilde{\vartheta}_{i_x},\ \tilde{\vartheta}_{i_y}\}$$

The hierarchical indices for $\tilde{\vartheta}_{i_x}$ and $\tilde{\vartheta}_{i_y}$ are, respectively, $\{k_{x_i},\ j_{x_i}\}$ and $\{k_{y_i},\ j_{y_i}\}$.
Furthermore, we suppose that, without loss of generality, this patch is a rectangle whose width is 2^{-j_H} and height 2^{-j_V}, with $j_H,\ j_V \in \mathbb{N}$.
Denoting now the refined patch vertices as $\mathcal{V}_{r_1}$, $\mathcal{V}_{r_2}$, $\mathcal{V}_{r_3}$, $\mathcal{V}_{r_4}$, $\mathcal{V}_{r_5}$, placed as in Figure 6.4, in the following we derive, starting from the vertices of the patch we are refining, the hierarchical indices $k^{(r)}_{x_i}$, $j^{(r)}_{x_i}$, $k^{(r)}_{y_i}$, $j^{(r)}_{y_i}$ for each sub-patch vertex:

$$\mathcal{V}_{r_1} = \{\tilde{\vartheta}_{1_x},\ \tilde{\vartheta}_{1_y}\}, \qquad \begin{aligned} k^{(r)}_{x_1} &= k_{x_1} \cdot 2^{j_H+1-j_{x_1}} + 1 \\ j^{(r)}_{x_1} &= j_H + 1 \\ k^{(r)}_{y_1} &= k_{y_1} \\ j^{(r)}_{y_1} &= j_{x_1} \end{aligned} \tag{6.2}$$

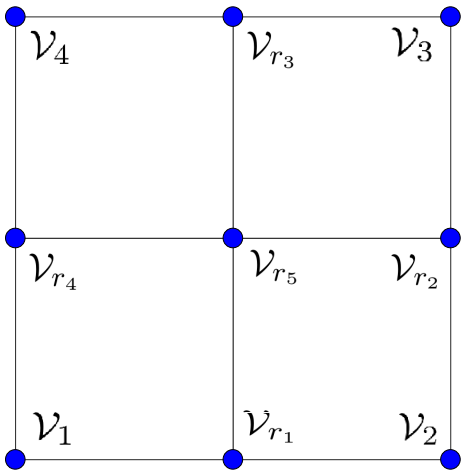


Figure 6.4: Ordering of refined vertices

$$\mathcal{V}_{r_2} = \{\tilde{\vartheta}_{2_x},\ \tilde{\vartheta}_{2_y}\},$$

$$k_{x_2}^{(r)} = k_{x_2}$$
$$j_{x_2}^{(r)} = j_{x_2}$$
$$k_{y_2}^{(r)} = k_{y_2} \cdot 2^{j_V+1-j_{y_2}} + 1$$
$$j_{y_2}^{(r)} = j_V + 1$$

$$\mathcal{V}_{r_3} = \{\tilde{\vartheta}_{3_x},\ \tilde{\vartheta}_{3_y}\},$$

$$k_{x_3}^{(r)} = k_{x_3} \cdot 2^{j_H+1-j_{x_3}} - 1$$
$$j_{x_3}^{(r)} = j_H + 1$$
$$k_{y_3}^{(r)} = k_{y_3}$$
$$j_{y_3}^{(r)} = j_{y_3}$$

$$\mathcal{V}_{r_4} = \{\tilde{\vartheta}_{4_x},\ \tilde{\vartheta}_{4_y}\},$$

$$k_{x_4}^{(r)} = k_{x_4}$$
$$j_{x_4}^{(r)} = j_{x_4}$$
$$k_{y_4}^{(r)} = k_{y_4} \cdot 2^{j_V+1-j_{y_4}} - 1$$
$$j_{y_4}^{(r)} = j_V + 1$$

$$\mathcal{V}_{r_5} = \{\tilde{\vartheta}_{5_x},\ \tilde{\vartheta}_{5_y}\},$$

$$k_{x_5}^{(r)} = k_{x_1} \cdot 2^{j_H+1-j_{x_1}} + 1$$
$$j_{x_5}^{(r)} = j_H + 1$$
$$k_{y_5}^{(r)} = k_{y_2} \cdot 2^{j_V+1-j_{y_2}} + 1$$
$$j_{y_5}^{(r)} = j_V + 1$$

These equations allow us to retrieve sub-patches vertices coordinates, thus enabling to completely define them. Thus, according to the previous categorization, in refining a patch the following rules hold:

- To perform a full-refinement we need to compute all the vertices coordinates $\mathcal{V}_{r_1}$, $\mathcal{V}_{r_2}$, $\mathcal{V}_{r_3}$, $\mathcal{V}_{r_4}$, $\mathcal{V}_{r_5}$
- To perform a vertical refinement we need to compute $\mathcal{V}_{r_2}$ and $\mathcal{V}_{r_4}$
- To perform a horizontal refinement we need to compute $\mathcal{V}_{r_1}$ and $\mathcal{V}_{r_3}$

The need of three different sub-patches types comes along with the necessity of improving the efficiency of the adaptive algorithm. In principle, to avoid losing any violation area, a full patch refinement would be necessary. However, there are cases in which this strategy turns out to be too conservative and unnecessarily time-consuming. For this reason we introduced the other two methodologies, enabling to save computational time (they require to sample just two points instead of five) in these cases where a full-refinement does not add useful information with respect to the other ones.

Summarizing, in this section we discussed about a 2-dimensional parameter space and its normalization. Moreover, we introduced a parameter space tessellation strategy to be used in localizing passivity violations. In the following section we will present in detail the adaptive refinement technique used in a bi-variate passivity verification algorithm.

6.3 Bi-Dimensional Adaptive Refinement Scheme

In this section we are going to present an adaptive bi-dimensional sampling scheme for bi-variate macro-models passivity verification. The main framework remains unchanged with respect to the mono-parametric case. In fact, an initial uniform sampling, whose aim is to coarsely define violation macro-areas, is followed by a number of refined passes, if necessary, used to better characterize the more critical ones.

6.3.1 Coarse Sampling

The proposed passivity verification algorithm starts with a coarse uniform sampling of the parameter space $\widetilde{\Theta}$ by subdividing both directions into $2^{j_0^x}$ and $2^{j_0^y}$ intervals, for x and y directions respectively. To define the initial refinement levels j_0^x and j_0^y we proceed similarly to what we did in the mono-parametric case (see Section 5.2.1). However, since the heuristic rule in Equation (5.18) is strongly related to the cardinality of the parameter basis functions, the two refinement levels are, in general, different. Denoting with $\bar{\ell}_x$ and $\bar{\ell}_y$ the cardinalities of these basis functions for x and y parameters respectively, we can retrieve the initial refinement levels as shown in Equation (6.3):

$$\begin{aligned} j_0^x &= \log_2\lceil \kappa\ \bar{\ell}_x \rceil \\ j_0^y &= \log_2\lceil \kappa\ \bar{\ell}_y \rceil \end{aligned} \tag{6.3}$$

for $\kappa > 1$.

A number $2^{j_0^x} + 1$ and $2^{j_0^y} + 1$ of uniformly spaced points, stored respectively in $\mathcal{S}_0^x$ and $\mathcal{S}_0^y$, are thus identified along x and y directions. Then, the parameter space points to be sampled at this stage, denoted as $\mathcal{S}_G$, are defined by the cartesian product $\mathcal{S}_0^G = \mathcal{S}_0^x \times \mathcal{S}_0^y$. After all these points have been sampled, $2^{j_0^x + j_0^y}$ patches are constructed starting from the coordinates in $\mathcal{S}_0^G$. For us, constructing a patch, means that, starting from the vertices coordinates, we define its edges and we store the corresponding patch information as elementary cell in a data-structure. In the case of a coarse sampling, this data-structure is defined as $\mathcal{Q}^{j_0}$ and the i-th patch in it is denoted as $\mathcal{Q}_i^{j_0}$.
Figure 6.5 shows an example of the normalized parameter space $\widetilde{\Theta}$ after the initial sampling, performed with $j_0^x = 2$ and $j_0^y = 3$. Blue dots represent the sampled points, defined by the cartesian product, while the black lines denote the patch edges.

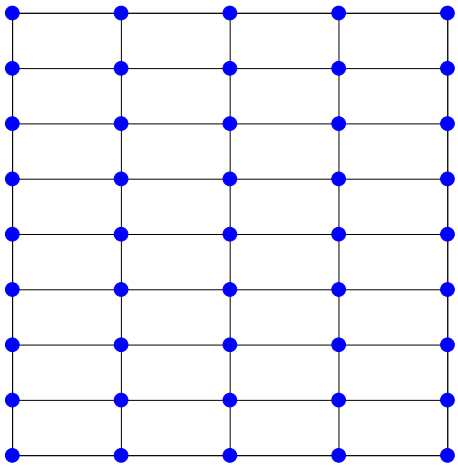


Figure 6.5: Initial coarse sampling in $\widetilde{\Theta}$ with $j_0^x = 2$ and $j_0^y = 3$

6.3.2 Refined Adaptive Sampling

In this section we are going to discuss about the adaptive refinement strategy.
In Chapter 4 and 5 we proposed two adaptive passivity assessment methodologies for mono-variate models. In the bi-variate case we chose to follow the derivative-based approach (Chapter 5), due to its higher reliability, as numerical results show in Section 5.4, and because it can be easily extended to a multi-variate case with minor modifications.

The mathematical framework we derived in Section 5.1 about Hamiltonian eigenvalues derivatives must be now extended to a multi-dimensional case.
To this end, given the eigen-spectrum $\Lambda_{\boldsymbol{\vartheta_0}}$ of a Hamiltonian matrix $\mathbf{M}(\boldsymbol{\vartheta_0})$ or a pencil $(\mathbf{M}(\boldsymbol{\vartheta_0}), \mathbf{K})$, we want to compute the set of linearly perturbed eigenvalues $\widehat{\Lambda}_{\boldsymbol{\vartheta_0}}$ at generic parameter space point $\boldsymbol{\vartheta} = \boldsymbol{\vartheta_0} + \Delta\boldsymbol{\vartheta}$, where $\boldsymbol{\vartheta_0} = [\vartheta_0^1, \ldots, \vartheta_0^\rho]$ and $\Delta\boldsymbol{\vartheta} = [\delta\vartheta^1, \ldots, \delta\vartheta^\rho]$.

Considering thus an eigenvalue $\lambda \in \Lambda_{\boldsymbol{\vartheta_0}}$, a variation $\delta\vartheta^i$ along the i-th parameter induces, at first order, an eigenvalue perturbation $\delta\lambda^i$ that reads:

$$\delta\lambda^i = \frac{\partial\lambda}{\partial\vartheta^i}|_{\boldsymbol{\vartheta}=\boldsymbol{\vartheta_0}} \cdot \delta\vartheta^i \tag{6.4}$$

To compute this derivative we must evaluate $\mathbf{M}(\boldsymbol{\vartheta_0})$ for all the parameters, except for the i-th one, and follow the same procedure detailed in Section 5.1.
Assuming that the perturbations $\delta\lambda^i$ are known for $i = 1, \ldots, \rho$, we can compute the perturbation of λ in $\boldsymbol{\vartheta}$, denoted with $\widehat{\lambda}$, as:

$$\widehat{\lambda} = \lambda + \sum_{i=1}^{\rho} \delta\lambda^i \tag{6.5}$$

In the case of a 2-dimensional parameter space, the terms $\boldsymbol{\vartheta_0}$ and $\Delta\boldsymbol{\vartheta}$ read:

$$\boldsymbol{\vartheta_0} = [\vartheta_0^{(1)},\ \vartheta_0^{(2)}] = [\vartheta_0^{(x)},\ \vartheta_0^{(y)}]$$

$$\Delta\boldsymbol{\vartheta} = [\delta\vartheta_0^{(1)},\ \delta\vartheta_0^{(2)}] = [\delta\vartheta_0^{(x)},\ \delta\vartheta_0^{(y)}]$$

Thus, the perturbed eigenvalue $\widehat{\lambda}$ is:

$$\widehat{\lambda} = \lambda + \delta\lambda^x + \delta\lambda^y$$

where

$$\delta\lambda^x = \frac{\partial\lambda}{\partial\vartheta^x}|_{\boldsymbol{\vartheta}=\boldsymbol{\vartheta_0}} \cdot \delta\vartheta^x$$

$$\delta\lambda^y = \frac{\partial\lambda}{\partial\vartheta^y}|_{\boldsymbol{\vartheta}=\boldsymbol{\vartheta_0}} \cdot \delta\vartheta^y$$

What is important to notice is that, to retrieve linearly approximated perturbed eigenvalues, it suffices to compute ρ mono-dimensional perturbations, as in Equation (6.4), and then combine them together as in (6.5). Thus, the same method we developed for the mono-dimensional case can be applied here without modifications.
Finally, as for the mono-dimensional case, we define a multi-dimensional perturbation operator as:

$$\widehat{\Lambda}_{\boldsymbol{\vartheta_0}}(\boldsymbol{\vartheta}) = \mathcal{P}_{M,K}^{\Delta\boldsymbol{\vartheta}}(\boldsymbol{\vartheta_0})$$

Once we detailed how Hamiltonian eigenvalues perturbations can be applied to a multi-dimensional space, we are ready to discuss about the adaptive scheme.
The parameter space patches subdivision described in the previous section turns out to be fundamental in this context. Well-defined patch structures allow us to exploit their

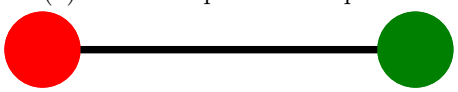
geometrical properties to detect if some critical areas (where a passivity violation may occur) are located inside them. In particular, we assume that, in order to decide whether the patch must be refined or not, just its edges have to be checked, leading us to easily select which type of refinement (full, vertical or horizontal) should be applied.
For this purpose, we identified the following rules:

- If a horizontal edge must be refined, a corresponding horizontal refinement is applied to the patch;
- If a vertical edge must be refined, a vertical refinement is applied to the patch;
- If both horizontal and vertical edges must be refined, then a full-refinement is applied to the patch.

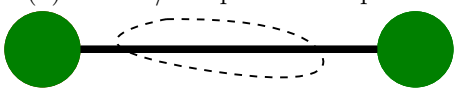
Therefore, the following strategy is used to detect if a generic patch edge needs a refinement

- If both edge end-points are not passive, a non-passive area around the edge has already been found and no additional samples are required. An example of this situation is shown in Figure 6.6a, where the red dots represent the non passive edge end-points;
- If one edge end-point is passive and the other is not, a violation area is located somewhere inside the edge. Thus, this edge is labelled as critical, since additional samples are then required to find the actual boundaries of this area. An example is given in Figure 6.6b, where the red and green dots represent the non-passive and passive end-points of the black edge;
- If both edge end-points are passive we can either infer that we identified a passive area or that a violation may be located between the two points. Then, to distinguish these two cases, a more refined check is needed. We present an example in Figure 6.6c, where the green dots are the passive edge end-points and the dashed line bounds a possibly non-passive area that, being not yet spotted, must be detected with a more refined technique.

(a) Both non-passive end-points



(b) Passive/non-passive end-points



(c) Passive end-points

Figure 6.6

To detect, in the third case, if some violation areas are likely to be located inside an edge aligned with the i-th parameter, we use the technique of Hamiltonian eigenvalues perturbations. Thus, we proceed by extracting all the Hamiltonian eigenvalues at both edge end-points, then we perturb them of a quantity $\delta\vartheta^i$ toward a testing node ϑ^i_T located at the edge mid-point (Figure 6.7 shows these perturbations applied on a generic patch edge). This double check is required to deal with the non-linearity of eigenvalues

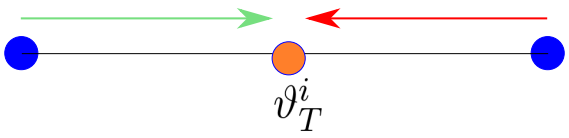


Figure 6.7: Perturbations directions on a generic patch edge

trajectories. Additionally, as detailed in Section 5.2.2, we note that a perturbation toward the edge center may be not enough to detect violations, specifically when strong non-linearities occur. Thus, we introduce a factor α, in such a way that we can modulate the perturbation extent as:

$$\left|\delta\vartheta^i_{ext}\right| = \alpha \cdot \left|\delta\vartheta^i\right|$$

In the majority of test-cases a value $\alpha = 1$ is enough to guarantee that all violation areas are spotted, but a suitable trade-off between computational times and reliability is given by α in the range $[\frac{3}{4},\ 2]$. In this bi-dimensional case the choice of this value greatly affects the CPU time, thus it must be chosen carefully.
We see that, with the assumption of looking just at patch edges, the bi-dimensional passivity verification problem reduces to the one we addressed in the mono-dimensional case, enabling to use the same strategy, illustrated in Table 5.1, with minor changes. Considering thus a 2^{nd} quadrant Hamiltonian eigenvalue $\lambda(\boldsymbol{\vartheta_0})$ evaluated for $\boldsymbol{\vartheta} = \boldsymbol{\vartheta_0}$ and its first order derivative with respect to the i-th parameter $\frac{\partial\lambda}{\partial\vartheta^i}|_{\boldsymbol{\vartheta}=\boldsymbol{\vartheta_0}}$, if $\widehat{\lambda}(\vartheta^i_0 + \delta\vartheta^i)$ denotes its perturbation induced by a parameter variation equal to $\delta\vartheta^i$, the rules in Table 6.1 hold (for a graphical representation of cases detailed below see Figures 5.7a–5.7d):

Case	$\mathrm{Re}\{\lambda(\boldsymbol{\vartheta_0})\} < -\delta_{th}$	$\mathrm{Re}\left\{\frac{\partial\lambda}{\partial\vartheta^i}|_{\boldsymbol{\vartheta}=\boldsymbol{\vartheta_0}}\right\} > 0$	Critical ?
1	NO	YES	YES
2	YES	YES	Check $\mathrm{Re}\left\{\widehat{\lambda}(\vartheta^i_0 + \delta\vartheta^i)\right\}$
3	YES	NO	NO
4	NO	NO	NO

Table 6.1: Multi-dimensional critical cases for adaptive refinement

As anticipated, we see that the definitions of critical cases have not changed with respect to the mono-dimensional case, they have been just extended to a higher dimension. Furthermore, we see that, in the context of a multi-dimensional space, the derivative-based framework enables to reduce the problem of detecting passivity violations to the computation of perturbed Hamiltonian eigen-spectra along suitable directions defined by

the perturbation vector $\Delta\vartheta$. Once these perturbed eigenvalues are known, the same rules in Table 6.1 hold.

So far, by assumption, we focused just on a generic patch edge. However this verification strategy applies without modifications to all the edges, as shown in Figure 6.8.

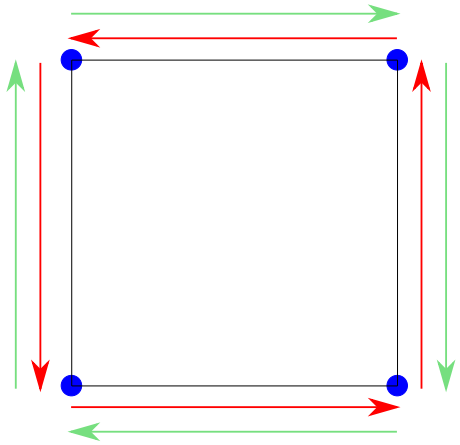


Figure 6.8: Patch edges eigenvalues perturbation directions

As one can see, the red and green arrows follow two different rotation directions. Thus, during the actual implementation, instead of focusing on single edges perturbations, it is better to execute all the clockwise followed by the counter-clockwise ones (or vice-versa). This arrangement enables us to have a cleaner implementation and better performances of the whole algorithm.

Now that we detailed how a patch edge is checked to detect possible nearby passivity violations, we can discuss about the strategy we pursue in checking, through its edges, the whole patch. Conversely from what we did in the mono-dimensional case, where as soon as a violation was detected a refinement was applied, here we store in a structure $\mathcal{R}$ the directions (vertical or horizontal) that are critical then, at the end of the verification process, a refinement is applied according to its content.
First of all, we coarsely characterize the patch through its vertices by detecting which are passive and which are not through a standard Hamiltonian-driven test. According to the results, the following cases may occur:

- If for all the vertices the model is not passive, no other verifications are required since we spotted a uniformly non-passive area. Thus, the patch is not refined and we proceed directly in analyzing the next one;
- If for all the vertices the model is passive, we need further verifications to characterize the patch. However, up to now, no refinement are needed, thus $\mathcal{R}$ is left empty;
- If the model is passive for some vertices and for other is not, we need to refine the patch according to the relative positions of passive and non-passive edges. Thus $\mathcal{R}$ is filled with the directions along which we need to refine.

At this stage, if not all the vertices are not passive, according to the cardinality of $\mathcal{R}$ (i.e. the number of critical directions it contains), we can directly proceed in fully refining the patch, if $\mathbf{card}(\mathcal{R}) = 2$, or we need to perform further verifications if $\mathbf{card}(\mathcal{R}) = 0$ or $\mathbf{card}(\mathcal{R}) = 1$.
If refined checks are required, we define a new structure $\bar{\mathcal{R}}$ that is, in some sense, complementary to $\mathcal{R}$, since it contains the directions that are not contained in $\mathcal{R}$, and that need additional tests to be labeled as critical or not. Thus, through the Hamiltonian eigenvalues perturbations technique, performed on the directions contained in $\bar{\mathcal{R}}$, we detect if some violations are likely to be found along the considered edges. If some direction is labeled as critical, it is added to $\mathcal{R}$.
Once all the required perturbations have been performed, we are ready to refine the patch according to the content of $\mathcal{R}$

- If $\mathcal{R} = \emptyset$, no refinement are needed;
- If it contains just the horizontal or vertical direction, the patch is refined accordingly;
- If it contains both directions, a full refinement is performed.

Once the the patch is refined and the sub-patches are created and stored in memory, we can remove from it the initial patch, since useless from now on.

The proposed adaptive refinement technique is cast inside an iterative loop that, at each iteration, checks just the elementary patches that have been refined at the previous one, stored in a data-structure $\mathcal{Q}^j$ with j an iteration index, avoiding expensive yet useless eigenvalues extractions on areas already characterized as passive or not passive. This iterative process stops when no refinements have been performed at the current iteration, which means that no other information is needed to characterize the model passivity, or when a maximum refinement level j_{max} is reached.

As result of this bi-variate passivity verification algorithm we obtain the locations of passivity violations. In particular, planning to use this verification scheme inside a multivariate extension of the iterative passivity enforcement algorithm presented in 4.4 (based on local linearized passivity constraints), we must locate the worst case violations (minimum eigenvalue/maximum singular value) in each non-passive frequency band, for a fixed parameters values combination $\boldsymbol{\vartheta}_{\mathbf{m}} = \{\vartheta_{m_x}, \vartheta_{m_y}\}$. Thus, to properly formulate passivity constraints we must store the largest violations as triplets $(\bar{\omega}_i, \boldsymbol{\vartheta}_{\mathbf{m}}, \underline{\lambda}_i)$ for immittance systems or $(\bar{\omega}_i, \boldsymbol{\vartheta}_{\mathbf{m}}, \underline{\sigma}_i)$ for scattering systems, where $\underline{\lambda}_i$ and $\underline{\sigma}_i$ denote the minimum eigenvalue of $\mathbf{H}(\mathrm{j}\bar{\omega}_i; \boldsymbol{\vartheta}_{\mathbf{m}}) + \mathbf{H}^{\mathsf{H}}(\mathrm{j}\bar{\omega}_i; \boldsymbol{\vartheta}_{\mathbf{m}})$ and the maximum singular value of $\mathbf{H}(\mathrm{j}\bar{\omega}_i; \boldsymbol{\vartheta}_{\mathbf{m}})$, respectively. All the passivity locations are collected in a data-structure $\mathcal{W}$.

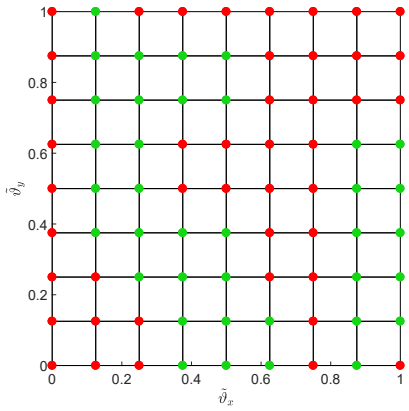
In Algorithm 6.11 we show a pseudo-code for the proposed verification scheme.

Algorithm 6.11 Bi-dimensional adaptive passivity verification algorithm

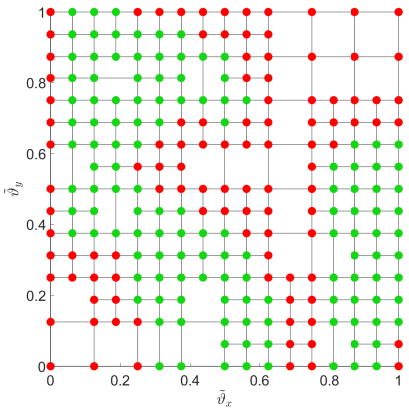
Require: Parametric model $\mathbf{H}(s;\boldsymbol{\vartheta})$
Require: Parameters basis orders $\bar{\ell}_x,\ \bar{\ell}_y$
Require: Algorithm control parameters $\kappa,\ \alpha,\ j_{max}$
Get initial refinement levels $j_0^x,\ j_0^y$ as in (6.3)
Sample initial points along x and y directions and store them in $\mathcal{S}_x,\ \mathcal{S}_y$ respectively
Get initial grid points $\mathcal{S}_0^G$ as $\mathcal{S}_0^x \times \mathcal{S}_0^y$
Sample points in $\mathcal{S}_0^G$ and store violations in $\mathcal{W}$ as $(\bar{\omega}_i, \boldsymbol{\vartheta}_\mathbf{m}, \underline{\lambda}_i)$ or $(\bar{\omega}_i, \boldsymbol{\vartheta}_\mathbf{m}, \underline{\sigma}_i)$
for $j = \max\{j_0^x, j_0^y\}, \ldots, j_{max}$ **do**
    Define the set of patches $\mathcal{Q}^j$ to be checked at the current iteration
    if $\mathcal{Q}^j = \emptyset$ **then**
        Return passivity violations $\mathcal{W}$
        break
    end if
    Set q as the cardinality of $\mathcal{Q}^j$
    for $i = 1, \ldots, q$ **do**
        Initialize $\mathcal{R} = \emptyset$
        Get vertices coordinates $\mathcal{V}_1, \ldots, \mathcal{V}_4$ of the i-th patch in $\mathcal{Q}^j$, denoted as $\mathcal{Q}_i^j$
        Check if $\mathbf{H}(s;\boldsymbol{\vartheta}_\mathbf{m})$ is passive in $\mathcal{V}_1, \ldots, \mathcal{V}_4$
        if $\mathbf{H}(s;\boldsymbol{\vartheta}_\mathbf{m})$ not passive $\forall\ \mathcal{V}_1, \ldots, \mathcal{V}_4$ **then**
            break
        else
            Store in $\mathcal{R}$ the directions to be refined, according to the adaptive strategy
        end if
        if cardinality of $\mathcal{R} = 2$ **then**
            Compute vertices coordinates $\mathcal{V}_{r_1}, \ldots, \mathcal{V}_{r_5}$ as in (6.2)
            Sample the vertices and store violations as $(\bar{\omega}_i, \boldsymbol{\vartheta}_\mathbf{m}, \underline{\lambda}_i)$ or $(\bar{\omega}_i, \boldsymbol{\vartheta}_\mathbf{m}, \underline{\sigma}_i)$ in $\mathcal{W}$
            Construct new sub-patches and store them in $\mathcal{Q}^{j+1}$
            break
        end if
        Set $\bar{\mathcal{R}}$ as the complementary of $\mathcal{R}$
        Get perturbed eigen-spectra $\widehat{\Lambda}_{\boldsymbol{\vartheta}_\mathbf{i}}(\vartheta) = \mathcal{P}_{M,K}^{\Delta\vartheta}(\boldsymbol{\vartheta}_\mathbf{i})$ centered in $\mathcal{V}_i$, along directions in $\bar{\mathcal{R}}$
        if $\widehat{\Lambda}_{\boldsymbol{\vartheta}_\mathbf{i}}(\vartheta)$ critical according to Table 6.1 **then**
            Update $\mathcal{R}$ with critical direction
        end if
    end for
    if $\mathcal{R} = \emptyset$ **then**
        break
    else
        Compute required vertices coordinates $\mathcal{V}_{r_1}, \ldots, \mathcal{V}_{r_5}$,as in (6.2), according to $\mathcal{R}$
        Sample the vertices and store violations as $(\bar{\omega}_i, \boldsymbol{\vartheta}_\mathbf{m}, \underline{\lambda}_i)$ or $(\bar{\omega}_i, \boldsymbol{\vartheta}_\mathbf{m}, \underline{\sigma}_i)$ in $\mathcal{W}$
        Create sub-patches and store them in $\mathcal{Q}^{j+1}$
    end if
end for
return Passivity violations $\mathcal{W}$

6.3.3 Numerical Results

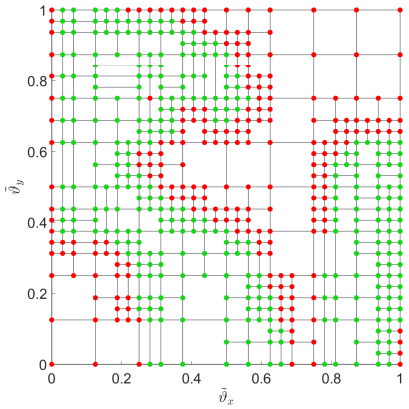
In this Section we are going to present some numerical results, showing the performances of the proposed verification scheme. In details, we want to focus on its ability to locate all the violation areas in the parameter space and refine just the more critical ones. To this end, we use an intermediate passivity enforcement model (fifth iteration) related to *Test Case 15* (for details see Appendix A), adaptively sampled with the proposed method. Figures 6.9a–6.9d represent the violations in parameter space related to four successive refinement iterations: green dots denote parameter space points where the model is passive $\forall\omega$ while the red-dots represent, instead, points in which the model is not passive for at least one frequency value.



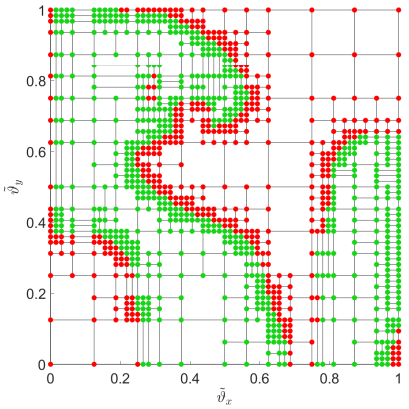
(a) Adaptive refinement: Iteration 1



(b) Adaptive refinement: Iteration 2



(c) Adaptive refinement: Iteration 3



(d) Adaptive refinement: Iteration 4

Figure 6.9: Successive adaptive refinement iterations) *Test Case 15*

We see that the proposed adaptive approach, whose settings are listed below:

- $j_0^x = 3$, $j_0^y = 3$
- $j_{max} = 6$

- $\alpha = \frac{3}{4}$

is able to iteratively locate and define all the violations areas in the parameter space and it is remarkable its ability to define passive/non-passive interface regions. In this respect, the choice of enabling the use of rectangular sub-patches turns out to be fundamental.
To verify the reliability of this method, we compare the result shown in Figure 6.9d with the one Figure 6.10, that represents the outcome of a fine uniform passivity check. We notice that the proposed strategy, precisely because of the look-ahead properties of

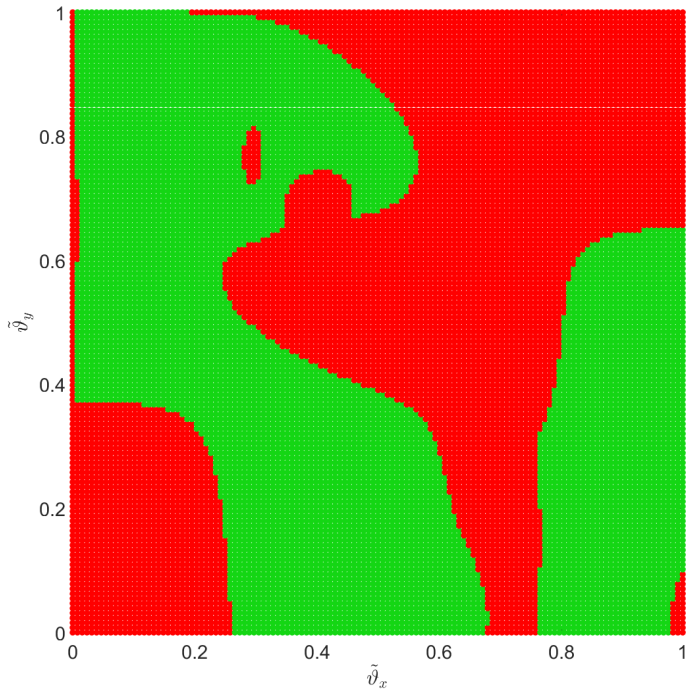


Figure 6.10: Uniform passivity verification outcome

Hamiltonian eigenvalue perturbations, is able to spot all the violation areas, even the smallest, as the one located in the upper-left quadrant. The difference in computational times is remarkable. In fact, the adaptive algorithm took 40 sec to reach the presented results, while the uniform check took around 2 minutes. This difference justifies the choice of an adaptive approach in parametric passivity verification.

6.4 Multi-Parametric Passivity Enforcement

In this section we will extend the local passivity enforcement algorithm shown in Section 4.4 to the multi-dimensional case. Before detailing how this can be done, we must discuss about the notation, in order to keep it compact even when the number of parameters increases. To this end, in the following, we will go through two different indexing methodologies for tensors, namely, the subscripts and the linear indexing.

6.4.1 Linear and Subscript Indexing for Tensors

Suppose that $\mathcal{T}_{i,j,k}$ is a 3-way tensor, whose entries are identified by the values assumed by indices i, j, k. Our aim is to vectorize the elements of this tensor through a suitable operator, similarly to what the operator $\mathbf{vec}(\cdot)$ does on matrices. Thus, assuming that $i = 1, \ldots, \bar{\imath}$, $j = 1, \ldots, \bar{\jmath}$, $k = 1, \ldots, \bar{k}$ and denoting as $\mathbf{T}_k$ the k-th slice extracted from the tensor along the third dimension, we define the operator $\mathbf{Vec}(\cdot)$ that, acting on $\mathcal{T}$ as shown in (6.6), returns its vectorization, denoted as $\mathbf{t}$:

$$\mathbf{t} = \mathbf{Vec}(\mathcal{T}) \;\Leftrightarrow\; \mathbf{t} = [\mathbf{vec}(\mathbf{T}_1)^\mathsf{T}, \ldots, \mathbf{vec}(\mathbf{T}_k)^\mathsf{T}, \ldots, \mathbf{vec}(\mathbf{T}_{\bar{k}})^\mathsf{T}]^\mathsf{T} \tag{6.6}$$

For a generic g-way tensor $\mathcal{T}$, the operator $\mathbf{Vec}(\cdot)$ re-shapes its elements according to the ordering followed by MATLAB built-in function *sub2ind*.

The same re-shaping operation can be performed on the indices of tensor elements. It is common in dealing with these objects to pass from a so-called subscript notation (the one with i, j, k indices) to a linear one, where each element belonging to the tensor is identified by a linear index. To better explain this operation we start with a bi-dimensional example. Denoting with $\mathbf{M}$ a 3×3 matrix, we identify its entries in subscripts notation as tuples (i, j). Our aim is to assign to each element in $\mathbf{M}$ a linear index that identifies it uniquely. In Figure 6.11, the left panel represents the subscripts notation while in the right panel is shown the linear one.

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

(a) Subscript indexing

1	4	7
2	5	8
3	6	9

(b) Linear indexing

Figure 6.11: Subscripts and linear indexing for a 3×3 matrix

Supposing now that $\mathbf{M}$ is a generic $\bar{m}_x \times \bar{m}_y$ matrix, whose elements are identified in

subscripts notation as $\mathbf{m} = (m_x,\ m_y)$, we can define a linear index $p = 1, \ldots, \bar{p}$, with $\bar{p} = \bar{m}_x \cdot \bar{m}_y$, that can be used to uniquely identify each matrix element. The operation that allows to pass from $\mathbf{m}$ to p is bijective and, for a two-dimensional case, is:

$$p = m_x + (m_y - 1)\bar{m}_x$$

We define thus an operator $\mathbf{IVec}(\cdot)$ that returns the linear index p, given the set of subscript indices $(m_x,\ m_y)$:

$$p = \mathbf{IVec}(\mathbf{m})$$

Thanks to bijectivity, we can retrieve subscripts indices starting from the linear one as:

$$m_x = ((p-1)\ \mathbf{mod}\ \bar{m}_x) + 1$$
$$m_y = \left\lfloor \frac{p-1}{\bar{m}_x} \right\rfloor + 1$$

In higher dimensions the relation that links the subscript indices of a tensor $\mathcal{M}$ with the linear one, follows the same ordering of the MATLAB built-in function *sub2ind*.
The one-to-one relation that links linear indices with subscripts enables us to identify tensor element just with a scalar. This, as we will see later on, turns out to be fundamental to compactly construct cost-function and constraints.

Some additional remarks about the notation are in order. In fact, we know that, in a model dependent on ρ external parameters, the variability that they induce is translated as the product of ρ parameter basis functions. To keep the notation compact, denoting as $\boldsymbol{\ell} = [\ell_1, \ldots, \ell_i, \ldots, \ell_\rho]^\mathsf{T}$ a multi-index whose i-th entry must satisfy $1 \leq \ell_i \leq \bar{\ell}_i$ and identifying as $\boldsymbol{\vartheta}_\mathbf{m}$ a point in Θ, we define as:

$$\xi_{\boldsymbol{\ell}}(\boldsymbol{\vartheta}_\mathbf{m}) = \xi_{\ell_1}(\vartheta^{(1)}_{m_1}) \cdot \ldots \cdot \xi_{\ell_i}(\vartheta^{(i)}_{m_i}) \cdot \ldots \cdot \xi_{\ell_\rho}(\vartheta^{(\rho)}_{m_\rho})$$

the value of the parameters basis functions product evaluated at $\boldsymbol{\vartheta}_\mathbf{m}$. In this section we fixed some notation to deal with tensors. In the next one, we will extensively use it in defining matrices to be used in a multi-variate passivity enforcement algorithm.

6.4.2 Passivity Enforcement Algorithm

In this section we are going to extend the mono-dimensional passivity enforcement algorithm detailed in Section 4.4 to a higher dimensional case. To avoid repetitions, we report just the main differences that arise when dealing with multi-parametric models.
Supposing to use the residues perturbation technique, the perturbation matrix $\boldsymbol{\Delta}\mathbf{H}(s; \vartheta_\mathbf{m})$ is defined as in (4.5) and the decision variables are stored in $\mathbf{x}$ as:

$$\mathbf{x} = [\mathbf{x}^\mathsf{T}_{1,1}, \ldots, \mathbf{x}^\mathsf{T}_{i,j}, \ldots, \mathbf{x}^\mathsf{T}_{P,P}]^\mathsf{T}$$

where

$$\mathbf{x}_{i,j} = [(\boldsymbol{\Delta}\mathbf{R}_{0,1})_{i,j}, \ldots, (\boldsymbol{\Delta}\mathbf{R}_{n,\ell})_{i,j}, \ldots, (\boldsymbol{\Delta}\mathbf{R}_{\bar{n},\bar{\ell}})_{i,j}]^\mathsf{T}$$

where now ℓ is a multi-index.
Thus, the vector $\mathbf{x}$ stacks into a single column all elements of a high-order $(\rho + 3)$-way

tensor with dimensions $P \times P \times \bar{n} \times \bar{\ell}_1 \times \cdots \times \bar{\ell}_\rho$.

Recalling that the optimization problem we solve to find residues perturbation is:

$$\begin{aligned} &\min \|\Delta\mathbf{H}(s;\vartheta_{\mathbf{m}})\|^2 \\ &\text{s.t. } \mathbf{H}(s;\vartheta_{\mathbf{m}}) + \Delta\mathbf{H}(s;\vartheta_{\mathbf{m}}) \text{ is passive.} \end{aligned}$$

the first main difference arises in de-embedding the decision variables from the cost-function $\|\Delta\mathbf{H}(s,\vartheta_{\mathbf{m}})\|^2$. Following the procedure detailed in Section 4.4, we must generalize the definition of $\beta_{k,m;n,\ell}$ that, for a mono-dimensional model, a 4-way tensor. Firstly, being $\vartheta_{\mathbf{m}}$ a continuous variable in Θ, we must discretize it. Thus, supposing that the number of external parameters is ρ, we discretize the frequency-parameter space as a $(\rho+1)$-dimensional grid, whose vertices are identified by indices i along the frequency dimension and $(m_1,\ldots,m_\rho)$ along parameter axis; the latter are then collected in a multi-index $\mathbf{m}$. Thus, we denote as $\boldsymbol{\vartheta}_{\mathbf{m}}$ a vector containing the coordinates of a point in the discretized parameter space, identified by indices $\mathbf{m} = [m_1,\ldots,m_\rho]$.
Assuming that the number of frequency samples is $\bar{k}$ and that, along the i-th parameter, we have $\bar{m}_i$ discrete parameter samples, it holds that:

$$\beta \in \mathbb{R}^{(\bar{k}\times\bar{m}_1\times\cdots\times\bar{m}_\rho)\times(\bar{n}\times\bar{\ell}_1\times\cdots\times\bar{\ell}_\rho)}$$

Supposing now to extract from β the element related to frequency/parameter sample with indices $(k,\mathbf{m})$ and associated to the parameter basis functions with orders $(n,\boldsymbol{\ell})$ we define:

$$(B_{i,j})_{p_r,p_c} := \beta_{k,\mathbf{m};n,\boldsymbol{\ell}} = \frac{\xi_{\boldsymbol{\ell}}(\boldsymbol{\vartheta}_{\mathbf{m}})\varphi_n(\mathrm{j}\omega_k)}{\mathsf{D}(\mathrm{j}\omega_k,\vartheta_{\mathbf{m}})}$$

where

$$\begin{aligned} p_r &= \mathbf{IVec}(k,\boldsymbol{m}) \\ p_c &= \mathbf{IVec}(n,\boldsymbol{\ell}) \end{aligned}$$

The matrix $\mathbf{B}_{i,j} \in \mathbb{R}^{\bar{p}_r\times\bar{p}_c}$ defined in Section 4.4 contains, with a suitable ordering, the terms $(B_{i,j})_{p_r,p_c}$ related to all the frequency/parameter samples, associated with all the basis function combinations. By construction, matrices $\mathbf{B}_{i,j}$ are equal $\forall i,j = 1,\ldots,P$: we use this notation to easily construct a compact cost-function, as shown in Equation (6.7).
Then, computing an economy-size QR-factorization of $\mathbf{B}_{i,j}$ as:

$$\mathbf{B}_{i,j} = \mathbf{Q}_{i,j}\ \boldsymbol{\Xi}_{i,j}, \quad \mathbf{Q}_{i,j}^{\mathsf{T}}\ \mathbf{Q}_{i,j} = \mathbb{I}$$

(see Section 4.4 for details), the term $\mathcal{E}_{i,j}^2$, defined in (4.7), can be written as:

$$\mathcal{E}_{i,j}^2 = \|\boldsymbol{\Xi}_{i,j}\mathbf{x}_{i,j}\|_2^2$$

Thus, the cost-function can be cast in the following compact form:

$$\begin{aligned} &\min \|\boldsymbol{\Xi}\ \mathbf{x}\|_2^2, \\ &\boldsymbol{\Xi} = \mathrm{blkdiag}\{\boldsymbol{\Xi}_{i,j}\}_{i,j=1}^{P} \end{aligned} \tag{6.7}$$

The other main difference from the mono-dimensional case is related to the formulation of linear passivity constraints. In Section 4.4.1, in order to explicit the decision variables, we introduced the tensor $\alpha_{i,\mu;n,\ell}$. In a higher dimensional case, we generalize this notation defining $\alpha_{n,\boldsymbol{\ell}}(\omega, \boldsymbol{\vartheta})$ as:

$$\alpha_{n,\boldsymbol{\ell}}(\omega, \boldsymbol{\vartheta}) = \frac{\xi_{\boldsymbol{\ell}}(\boldsymbol{\vartheta})\varphi_n(\mathrm{j}\omega)}{(\mathrm{j}\omega, \boldsymbol{\vartheta})}$$

with $\boldsymbol{\vartheta}$ a point in the discretized parameter domain. Additionally, we define an index z as:

$$z := (\omega,\ \boldsymbol{\vartheta})$$

thus, supposing we are formulating a constraint for $\omega = \omega_i$ and $\boldsymbol{\vartheta} = \boldsymbol{\vartheta}_{\boldsymbol{\mu}}$, with $\boldsymbol{\mu}$ a ρ dimensional multi-index, it holds that

$$\alpha_{n,\boldsymbol{\ell}}(z_j) = \alpha_{n,\boldsymbol{\ell}}(\omega_i, \boldsymbol{\vartheta}_{\boldsymbol{\mu}})$$

with $z_j = (\omega_i,\ \boldsymbol{\vartheta}_{\boldsymbol{\mu}})$.

Assuming that, for a given combination of basis functions orders $(n, \boldsymbol{\ell})$, a number $\bar{\imath}$ of constraints must be formulated, we define a matrix $\mathbf{A}_{n,\boldsymbol{\ell}}$ as

$$\mathbf{A}_{n,\boldsymbol{\ell}} = \begin{pmatrix} \alpha^{\mathsf{T}}_{n,\boldsymbol{\ell}}(z_1) \\ \vdots \\ \alpha^{\mathsf{T}}_{n,\boldsymbol{\ell}}(z_i) \\ \vdots \\ \alpha^{\mathsf{T}}_{n,\boldsymbol{\ell}}(z_{\bar{\imath}}) \end{pmatrix}$$

collecting all the terms needed to construct passivity constraints as detailed in Equations (4.13), (4.18) for immittance and scattering systems, respectively.

Finally, once cost-function and constraints are properly formulated, the enforcement algorithm structure derived in the mono-dimensional case can be exploited as is without modifications also in this more general framework.

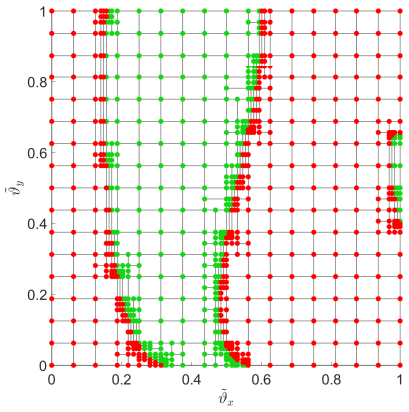
6.4.3 Numerical Results

In this section we are going to present some numerical examples showing the capabilities of the proposed multi-parametric passivity enforcement algorithm, combined with the bi-dimensional verification scheme detailed in the previous sections.
In details, we want to show that, given an initially non-passive model, the algorithm is able to reduce, iteration after iteration, the passivity violations, until a passive model is obtained.

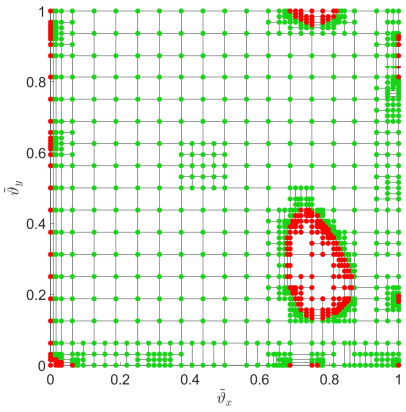
The first example is based on the *Test Case 16*, a high-speed PCB link (see Appendix A for details). We start by presenting an initially non-passive model, whose violations in the parameter space are represented in Figure 6.12a.
By setting the bi-dimensional passivity verification algorithm with the following parameters:

- $j_0^x = 4$, $j_0^y = 4$
- $j_{max} = 7$
- $\alpha = \frac{3}{4}$

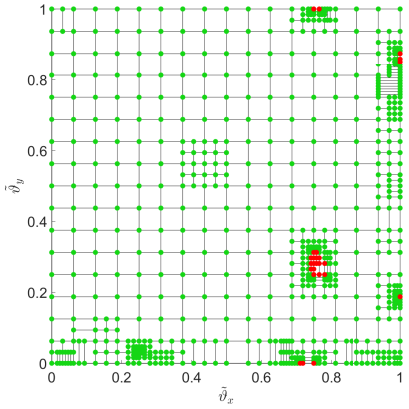
the proposed enforcement algorithm is capable of make it passive in 5 iterations. Figures 6.12a–6.12d show how passivity violations on the parameter space are reduced iteration after iteration, until a passive model, whose violations in the parameter space are shown in Figure 6.12d, is obtained.



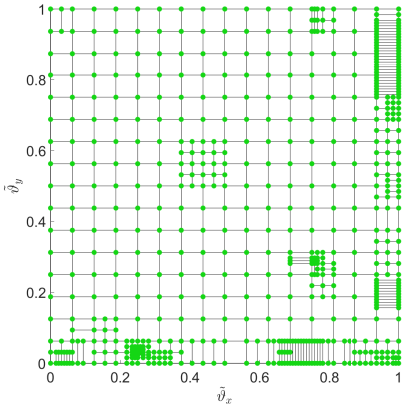
(a) Parameters space of the initial non-passive model



(b) Passivity enforcement iteration 1



(c) Passivity enforcement iteration 2



(d) Passivity enforcement final iteration

Figure 6.12: Successive bi-variate passivity enforcement iterations (*Test case 16*)

To be sure that a passive model is reached, we perform on it a further passivity verification on a uniform fine grid, with a refinement level $j_{max} = 7$, to be compliant with the settings of the adaptive verification algorithm. The result of this latter test, shown in Figure 6.13, clearly states that no violations are found in the parameter space, thus the

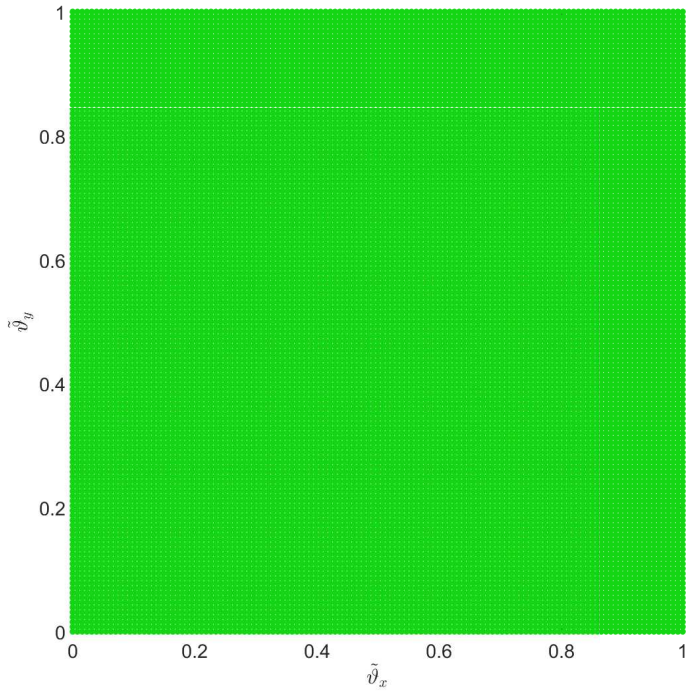
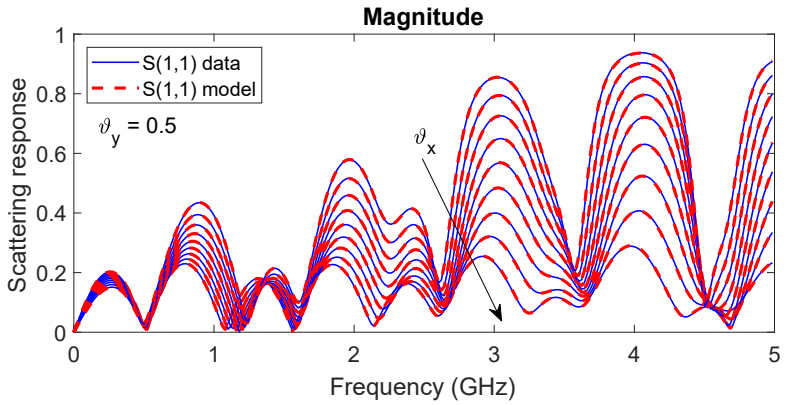


Figure 6.13: Uniform passivity verification outcome (*Test case 16*)

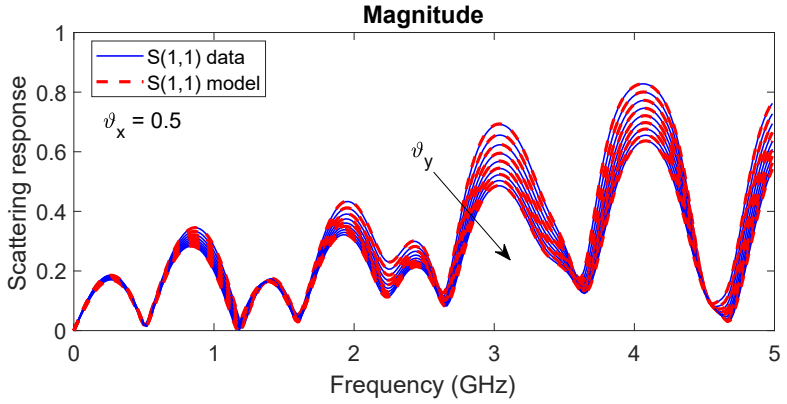
model is passive.

Once the guaranteed passive model has been obtained, we must be sure that its accuracy with respect to raw data is still acceptable. For this purpose, we compare the frequency response of ports (1, 1) of the passive model with respect to raw data for various parameters combinations. In particular, in Figure 6.14a we show the model response for $\tilde{\vartheta}_y$ fixed at 0.5 and varying $\tilde{\vartheta}_x$, while in Figure 6.14b we show the response for $\tilde{\vartheta}_x = 0.5$ and varying $\tilde{\vartheta}_y$. We see that, for every parameter combination, the model response is very accurate. The maximum relative error among all the ports combinations and for all parameters is equal to $14 \cdot 10^{-3}$, that is almost identical to the error of the non-passive model, showing that this enforcement procedure is able to keep the model accurate while imposing passivity.

About computational times, in the proposed example we chose as maximum refinement level $j_{max} = 7$, that leads the algorithm to complete the passivity enforcement in more or less 12 minutes, where around 5000 Hamiltonian eigenvalue extractions were necessary. We would like to emphasize that just the uniform sampling performed to obtain Figure 6.13 took 15 minutes, more than the time required by the adaptive algorithm to perform a complete enforcement. This result is another evidence of the fact that, especially in higher dimensions, the use of adaptive algorithm is fundamental.



(a)



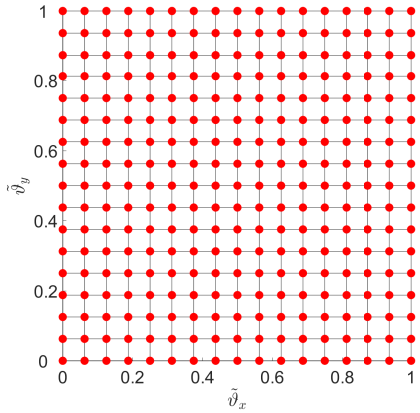
(b)

Figure 6.14: Uniform passivity verification outcome (*Test case 16*)

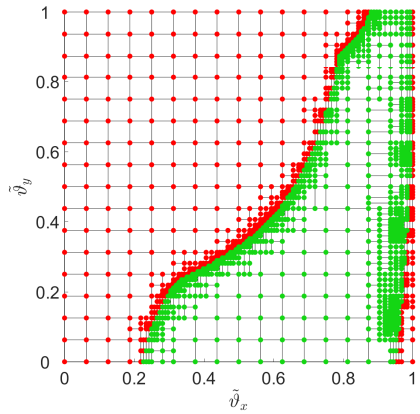
The second example we propose is an integrated inductor whose details are reported in the Appendix A, under *Test Case 17*. We show this test-case mainly because, being inductors strongly reactive circuit elements, it is likely that, due to the low intrinsic dissipativity, many violations may occur. An evidence of that is given in Figure 6.15a, that shows the violations on the parameter space for the initial non-passive model: we see that the model is not passive for all the parameter combinations in the uniform grid.
By setting verification algorithm as detailed below:

- $j_0^x = 4$, $j_0^y = 4$
- $j_{max} = 7$
- $\alpha = 2$

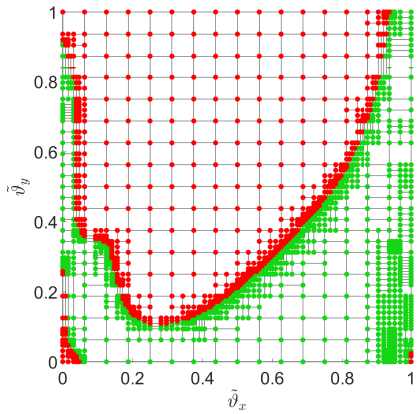
the enforcement algorithm is able to return a passive model, whose violations in the parameter space are shown in Figure 6.15f, in 12 iterations.
Figures 6.15a–6.15f represent violations in the parameter space for 6 enforcement iterations. Again, we notice how the violation areas are well-spotted and removed by the



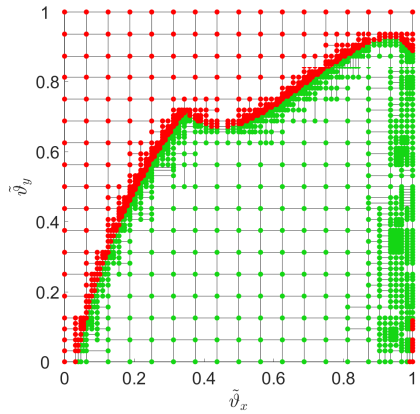
(a) Parameters space of the initial non-passive model



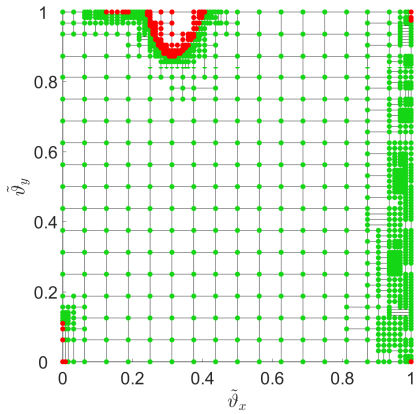
(b) Passivity enforcement iteration 4



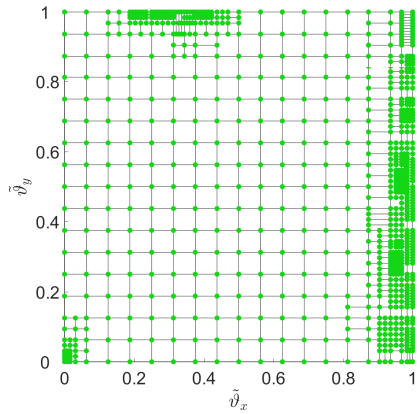
(c) Passivity enforcement iteration 5



(d) Passivity enforcement iteration 6



(e) Passivity enforcement iteration 8



(f) Passivity enforcement final iteration

Figure 6.15: Successive bi-variate passivity enforcement iterations for *Test Case 17*

verification and enforcement algorithms.
To be sure that the obtained model is passive we perform a uniform passivity check on a fine parameter grid with refinement level $j_{max} = 7$, to be compliant with the verification algorithm settings. Figure 6.16 shows violations in the parameter space of the candidate passive model: we see that no violations are present, thus we are sure that the resulting model is passive.

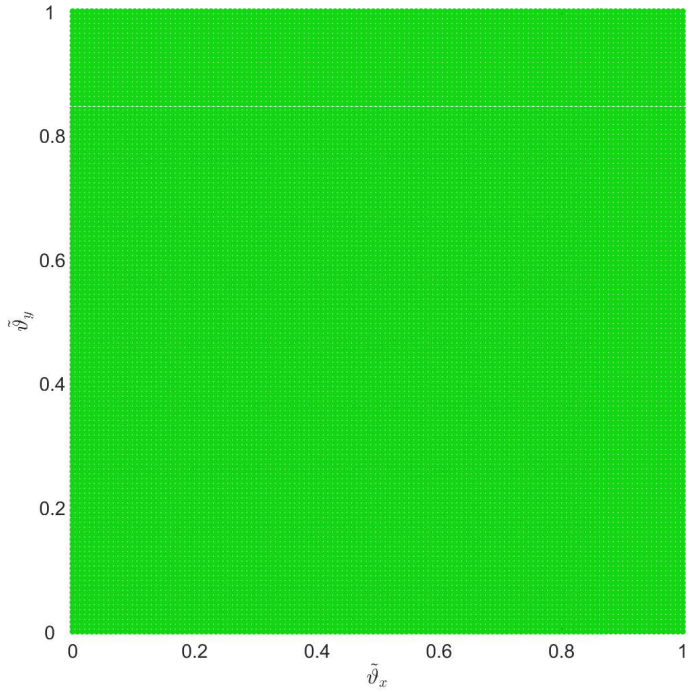


Figure 6.16: Uniform passivity verification outcome for *Test case 17*

In this Chapter we introduced the passivity verification and enforcement problem for multi-variate macro-models. In the introductory part we stated that for high-dimensionality models, fundamental issues related to computational complexity arise. The proposed adaptive algorithm tries to overcome, to some extent, these problems for bi-dimensional models. However, by comparing the results we obtained in the mono-dimensional case, particularly in terms of computational times, we see that the latter have considerably increased.
A three-parameter case could be the natural extension of the proposed algorithm, however we argue that CPU times would no longer be manageable. In the next, and final, Chapter we will briefly introduce and discuss some advanced techniques that should enable us to deal with high-dimensional parameters spaces.

Conclusions

In this work, we discussed passivity verification and enforcement methods for macro-models, with emphasis on parametric and multi-parametric structures. For the latter, we proposed an innovative passivity verification scheme for mono- and bi-variate macro-models, for which theoretical and practical implementation aspects are discussed in details. Furthermore, several numerical results show the reliability and the efficiency of the proposed algorithms.
In the first Chapters, we detailed what macro-models are, discussing their structure and why they are so important in many engineering fields. In this scope, we pointed out that standard fitting algorithms (Generalized Sanathanan-Koerner [35], [37], [10], [33]) and even the most advanced ones, such as the Fast Parameterized Sanathanan-Koerner [19], [5], are not able to extract guaranteed passive macro-models thus, an a-posteriori passivity enforcement procedure is required.
We started addressing the problem of non-parametric passivity verification and enforcement in Chapter 3, where we introduced an algebraic method, the Hamiltonian-driven technique [17, 21], to spot passivity violation areas along the frequency axis. Moreover, we discussed a passivity enforcement scheme, originally presented in [21] that, suitably perturbing model numerator coefficients, is capable to return a passive model starting from a non-passive one, without affecting its accuracy with respect to raw data.
Being parameterized macro-models our main focus, in Chapter 4 we introduced the problem of parametric macro-models passivity. In particular, we discussed an adaptive verification and enforcement scheme, originally presented in [18], that, exploiting the Hamiltonian-based frequency sampling, enables to efficiently check the model passivity on a bi-dimensional frequency-parameter space.
The main contribution of this work, presented in Chapters 5 and 6, aims to be an extension of state-of-the-art passivity assessment methodologies. In Chapter 5, we proposed a novel mono-parametric passivity verification method that, relying on the 'predictive' aspect of Hamiltonian eigenvalue perturbations, solves some reliability issues encountered with the method presented in Chapter 4. Additionally, after an in-depth discussion on asymptotic passivity characterization, we propose an innovative method to algebraically detect asymptotic passivity violations.
The several numerical results show that the proposed scheme effectively solves the mentioned reliability problems, without affecting computational times, that are in many cases even lower. This result is of paramount importance, because model labeled as passive by previous verification algorithms could present some residual violations, leading to possible simulation instabilities.

The same verification methodology is then exploited in Chapter 6 to develop an adaptive bi-variate passivity assessment scheme [47]. The proposed numerical examples state that our bi-dimensional adaptive strategy is very effective in locating passivity violation areas in a bi-dimensional parameter space, keeping reasonable the computational times. Furthermore, the parametric enforcement algorithm presented in Chapter 4 has been extended to a multi-dimensional case.
The realization of reliable mono- and bi-variate passivity verification algorithms turn out to be fundamental also in the field of stability enforcement. In fact, exploiting the techniques detailed in [8, 9, 22, 47], we are now able to enforce the model stability during the estimation process for parametric bi-variate macro-models.

Unfortunately, especially during the development of the bi-variate passivity verification algorithm, we realized that the presented adaptive approaches are un-feasible when the number of parameters increases, due to the *curse of dimensionality*. Thus, being conscious that many complex engineering problems are strongly multi-variate, further improvements, enabling to break the exponential dependence of computational requirements on the number of parameters, are in order. To this end, many advanced techniques, widely used in other scientific fields, are available, such as sparse-grids and tensor decompositions.
By means of sparse-grids, we refer to a set of numerical discretization techniques, extensively used in multi-variate problems, such as Finite Element Methods (FEM). These methods rely on the superposition (tensor product) of successive mono-dimensional basis grids. This approach enables to reduce the number of degrees of freedom, leading to a computational complexity that depends just logarithmically on the number of parameters. For details, see [6, 14].
Another way to efficiently deal with the curse of dimensionality in high dimensional problems is to exploit tensor properties. In fact, the most natural framework in dealing with high dimensional discrete grids are tensors. As reported in [48] and [42], tensor computation brings with it, by definition, the curse of dimensionality. However, with the help of decomposition techniques, such as high-order SVD and tensor trains, it is possible to represent large tensors through their decompositions, alleviating the issues induced by the possibly large number of dimensions.
Both methods are already well established in many scientific and engineering fields, such as PDE discretization and big-data analysis, thus we are confident that they can give a considerable performance boost to high-dimensional passivity verification schemes.

Appendix A

Test cases

In this Appendix, we collect and detail all the test cases used as numerical examples throughout this work. For each test case, we provide a brief description of the underlying physical structure as well as several bibliographic references, when available. Furthermore, being the model responses strongly related to fitting algorithm settings, such as basis functions orders, this information is also provided.

Remark: In the following, we will refer to models with ensured positive real denominator (see [8,22]) with the acronym *PR*, while the models with not ensured positive real denominator are identified as *not-PR*.

Case 1

Microstrip Filter with Double Folded Stub
The physical structure underlying this test case is a microstrip band-stop filter with double folded stubs, originally presented in [11] and [38] and depicted in Figure A.1

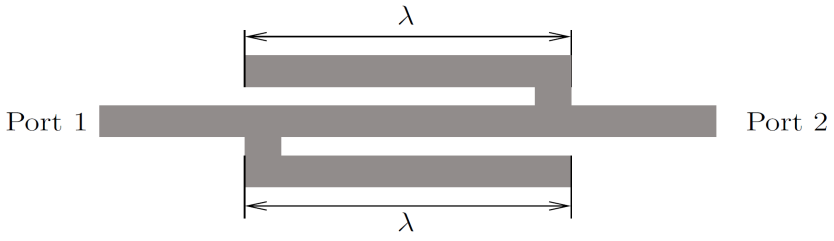


Figure A.1: Microstrip Filter with Double Folded Stub

We took as free parameter the length λ of the folded stubs, variable in [2.08, 2.28] mm. From successive simulations, we have 21 datasets containing frequency responses obtained for different parameter values. Each set contains $\bar{k} = 300$ frequency samples, with a bandwidth of [5, 20] GHz. To estimate the model we used 11 of them, while the other

ones are exploited for validation purposes. We extracted the model with partial fractions basis order $\bar{n} = 10$ and with Chebyshev polynomials as parameter basis functions, with orders $\bar{\ell}_N = \bar{\ell}_D = 2$ for numerator and denominator, respectively.
The maximum relative error between the PR-model and validation data is $4.5 \cdot 10^{-3}$, while for the not-PR model the error is $4.4 \cdot 10^{-3}$.

Case 2

Link on printed circuit board
This structure is an s-shaped link, 300 μm wide, placed on a printed circuit realized on a FR4 epoxy board with thickness 0.76 mm, dielectric constant equal to 4.4 and loss tangent 0.02. The structure, originally presented in [39], is depicted in Figure A.2. The free parameter is the middle-segment length, indicated with L in the Figure, variable in the range $[2,\ 18]$ mm, while the first and third segments have a fixed length of 2 cm.
The dataset is composed of 9 sets of frequency responses, obtained for different parameter values, composed themselves of $\bar{k} = 100$ frequency samples spanning the band $[0.1,\ 10]$ GHz.

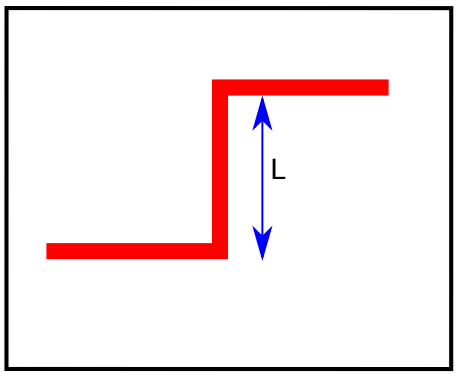


Figure A.2: Link on printed circuit board

Different models of this structure have been used throughout this work: the characteristics of each one are listed below.

Case 2–a

The model to which we refer here has partial fractions basis order $\bar{n} = 16$ and parameter basis functions (Chebyshev polynomials) orders $\bar{\ell}_N = 4$ and $\bar{\ell}_D = 3$ for numerator and denominator, respectively. This choice led us to use all the available datasets to fit the model.
The worst-case relative error for the PR-model with respect to fitting data is $200 \cdot 10^{-3}$ while, for the not-PR, is $54.9 \cdot 10^{-3}$.

Case 2–b

In this case, the model orders are $\bar{n} = 16$ and $\bar{\ell}_N = \bar{\ell}_D = 4$, where Chebyshev polynomials are used as parameter basis functions. To fit this model we used all the available datasets. With the orders listed above, and with 9 parametric datasets, the estimation problem is under-constrained along the parameter direction, leading to a possible loss of accuracy. However, for us this in not an issue, since this model has been extracted just to stress the presented passivity verification algorithms.
The largest relative error between the PR-model and raw fitting data is $1.5 \cdot 10^{-1}$, while for the not-PR model is $38.5 \cdot 10^{-3}$.

Case 3

Via with Residual Stub
This structure (depicted in Figure A.3), originally presented in [35], is a via connecting a microstrip line and a stripline in a multi-layer PCB. The metallization process that is performed to create the via, running from top to bottom, generates a residual stub, that is not necessary and may be a source of signal integrity problems. To reduce this issue, the stub height h can be adjusted through a backdrilling procedure. For this reason, we want to parameterize the model behavior with respect to the stub height h, in order to run simulations and find an optimal value for this parameter.

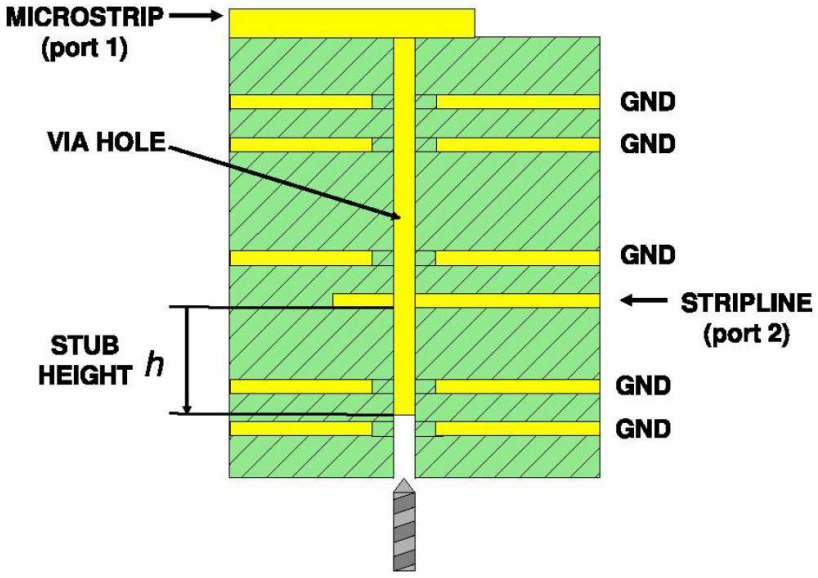


Figure A.3: Via with residual stub [35]

The dataset is composed of 10 sets of frequency responses obtained by means of a full-wave solver, simulating the structure behavior for 10 values of the stub height, assumed as free parameter, ranging from 0 μm to 716 μm. Each response is composed of $\bar{k} = 1001$

frequency samples, spanning the band [0, 40] GHz.
Different models of this structure have been used in this work, thus we list below the characteristics of each one.

Case 3–a

In this case, we extracted the model with partial fraction basis order $\bar{n} = 13$ and with Chebyshev polynomials as parameter basis functions, with orders $\bar{\ell}_N = 3$ and $\bar{\ell}_D = 3$ for numerator and denominator, respectively. Among the available parametric datasets, we used 6 of them to fit the model, while the other 4 are used for validation purposes.
The maximum relative error between the PR-model responses and validation data is $39.4 \cdot 10^{-3}$, while for the not-PR model the error is $41.5 \cdot 10^{-3}$.

Case 3–b

This model has been extracted with the same basis function orders as in *Test Case 3–a*. We changed the data used to fit the model: we use 5 parametric datasets for fitting and the remaining ones for validation.
The maximum relative error between the PR-model responses and validation data is $120 \cdot 10^{-3}$, while for the not-PR model is $84.7 \cdot 10^{-3}$.

Case 4

PCB Interconnect Over a Slotted Reference Plane
In this case, the structure is a microstrip disposed over a dielectric reference surface, where a rectangular discontinuity, placed at a distance d from the center of the reference plane and with length L, breaks the current return path [22]. The structure, depicted in Figure A.4

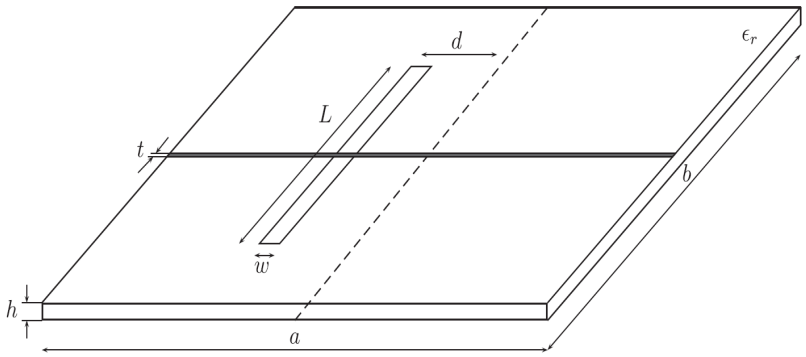


Figure A.4: PCB Interconnect Over a Slotted Reference Plane

has the following fixed geometrical parameters

- $a = 100$ mm;

- $b = 100$ mm;
- $\epsilon_r = 4.7$;
- $t = 0.035$ mm;
- $w = 0.12$ mm;
- $h = 0.3$ mm;
- $L = 25$ mm.

We consider as free parameters the slot length L, variable in [1, 25] mm, and its offset from the plane center d, which can vary in the range [0, 25] mm. Frequency responses were obtained with a full-wave solver for different parameter values combinations. Each dataset contains a set of $\bar{k} = 1858$ frequency samples, spanning the band [0, 10] GHz. These data were split to obtain smaller datasets, as detailed below.

Case 4–a

In this case, we fix the slot offset to 25 mm and consider as free parameter its length. To fit the model we used a partial fraction basis order $\bar{n} = 30$ and Chebyshev polynomials as parameter basis functions, with orders $\bar{\ell}_N = 6$ and $\bar{\ell}_D = 2$ for numerator and denominator, respectively. During the fitting process we used all the available parametric datasets. The worst-case relative error between the PR-model with respect to raw fitting data is $19.9 \cdot 10^{-3}$, while for the not-PR model we obtain an error of $9.3 \cdot 10^{-3}$.

Case 4–b

In this case, we fix the slot length to 25 mm and consider as free parameter the offset. To fit the model we used a partial fraction basis order $\bar{n} = 34$ and Chebyshev polynomials as parameter basis functions, with orders $\bar{\ell}_N = 10$ and $\bar{\ell}_D = 6$ for numerator and denominator, respectively. To extract the model we used all the available parametric datasets. We see that the chosen basis function orders are un-feasible with the number of available parametric datasets, leading the estimation problem to be under-constrained along the parameter dimension. However, this model has been extracted just to stress the proposed algorithms, thus we do not consider this as an issue.
The worst case relative error between the available data and the not-PR model is $2.5 \cdot 10^{-3}$.

Cases 5–8

Transmission Line With Embedded Discontinuity
This structure, originally presented in [22], is a transmission line with an embedded lumped RLC discontinuity, as depicted in Figure A.5.
We assume as free parameters the values of the discontinuity inductance L_1 and capacitance C. The values of the other components are fixed and listed below:

- Lines characteristic impedance $Z_\infty = 40\Omega$;

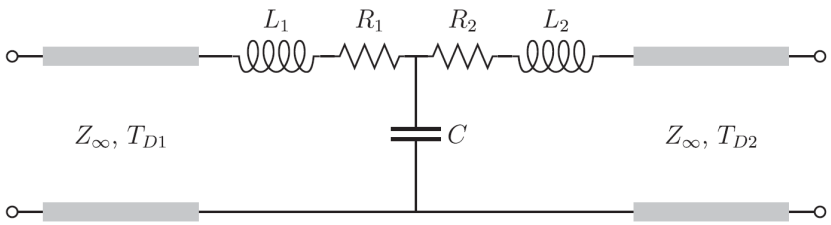


Figure A.5: Transmission Line With Embedded Discontinuity

- Propagation times $T_{D1} = 100$ ps, $T_{D2} = 230$ ps;
- Discontinuity resistance $R_1 = R_2 = 1\Omega$;
- Discontinuity inductance $L_2 = 10$ nH.

From repeated SPICE solver runs, we obtain 11 sets of parametric frequency responses, each one composed of $\bar{k} = 1000$ frequency samples, spanning the band [10 MHz, 10 GHz]. Depending on the values assumed by the free discontinuity parameters we have the following sub-cases.

Case 5

In this case, we assume as free parameter the capacitance C, that ranges in [0.1, 10] pF. The inductances $L_1 = L_2$ are fixed to 10 nH.
To extract the model we used a partial fractions order $\bar{n} = 18$ and Chebyshev polynomials as parameter basis functions, with orders $\bar{\ell}_N = \bar{\ell}_D = 1$. Six of the 11 available parametric datasets are used to fit the model, while the others are exploited for validation purposes. The largest relative error of the PR-model with respect to validation data is $4.6 \cdot 10^{-3}$, while for the not-PR model the error is $3.9 \cdot 10^{-3}$.

Case 6

In this case, we assume as free parameter the inductance L_1, that ranges in [10 pH, 1 nH], while the capacitance C is fixed to 1 pF. The model orders are $\bar{n} = 18$ for partial fraction basis and $\bar{\ell}_N = \bar{\ell}_D = 1$ for the parameter basis functions (Chebyshev polynomials). Six of the 11 available parametric datasets are used to fit the model, while the others are exploited for validation purposes.
The largest relative error of the PR-model with respect to validation data is $2.5 \cdot 10^{-3}$, while for the not-PR model the error is $2.1 \cdot 10^{-3}$.

Case 7

In this case, we assume as free parameter the capacitance C, that ranges in [0.1, 1] pF. The inductances $L_1 = L_2$ are fixed to 10 nH. The model has partial fractions basis order $\bar{n} = 18$ and Chebyshev polynomial as parameter basis functions, with orders $\bar{\ell}_N = 1$ and $\bar{\ell}_D = 2$. Five of the 11 available parametric datasets are used to fit the model, while the

other ones are exploited for validation purposes.
The largest relative error for both the PR and not-PR models with respect to validation data is $3.3 \cdot 10^{-3}$.

Case 8

In this case, we assume as free parameter the capacitance C, that ranges in [1, 10] pF. The inductances $L_1 = L_2$ are fixed to 10 nH. To extract the model we set the partial fractions order $\bar{n} = 18$ and Chebyshev polynomial as parameter basis functions, with orders $\bar{\ell}_N = 1$ and $\bar{\ell}_D = 1$. Six of the 11 available parametric datasets are used to fit the model, while the other ones are exploited for validation purposes.
The largest relative error for both the PR and not-PR models with respect to validation data is $0.768 \cdot 10^{-3}$.

Case 9

Multi-Layer integrated Inductor
In this case, the structure is a spiral integrated inductor with 1.5 turns, placed on a multilayer substrate (courtesy of Prof. M. Swaminathan, Georgia Institute of Technology, Atlanta, GA, USA), for details see [38] and [18]. The inductor has a square outline and its side-length is assumed as free parameter, with a range of variation in [1.02, 1.52] mm. From repeated simulations, we have a set of 11 frequency responses for different parameter values: 6 of them are used to fit the model, while the others are used for validation purposes. The model has been extracted with partial fractions basis order $\bar{n} = 8$ and with parameter basis function (Chebyshev polynomials) orders $\bar{\ell}_N = \bar{\ell}_D = 2$ for numerator and denominator, respectively.
The worst case relative error between the PR-model and validation data is $43.0 \cdot 10^{-3}$, while for the not-PR model the error is $3.6 \cdot 10^{-3}$.

Cases 10–11

Integrated Inductor
In this case, we refer to a PCB integrated inductor with a square outline, placed on a plane dielectric substrate, presented in [18, 38]. Depending on the number of turns it has (1.5 and 2), we have two different datasets. In both cases, the free parameter is the inductor side-length, that ranges in [1.02, 1.52] mm. From repeated simulations, we have 11 sets of frequency responses, each one containing $\bar{k} = 477$ frequency samples, spanning the band [0.1, 12] GHz. In the following, we will detail the model characteristics for the two cases.

Case 10

In this case, we refer to the inductor with 1.5 turns. The orders used to extract the model are $\bar{n} = 6$ for partial fractions and $\bar{\ell}_N = \bar{\ell}_D = 2$ for parameter basis functions (Chebyshev polynomials). All the available parametric datasets, except for one, were exploited as

fitting data.
The maximum relative error with respect to validation data is, for the PR-model $10.4 \cdot 10^{-3}$, while for the not-PR one is $0.783 \cdot 10^{-3}$.

Case 11

In this case, we refer to the inductor with 2 turns. The orders used to fit the model are the same as for *Test Case 10*, however, to extract it, we used just 6 parametric datasets, leading the others to be exploited for validation purposes.
The maximum relative error with respect to these validation data is, for the PR-model $19.9 \cdot 10^{-3}$, while for the not-PR one is $3.4 \cdot 10^{-3}$.

Cases 12–13

Coupled Transmission Lines
In this test case, we consider a set of N differential pairs, each formed by two parallel identical wires located one next to the other, as depicted in Figure A.6.

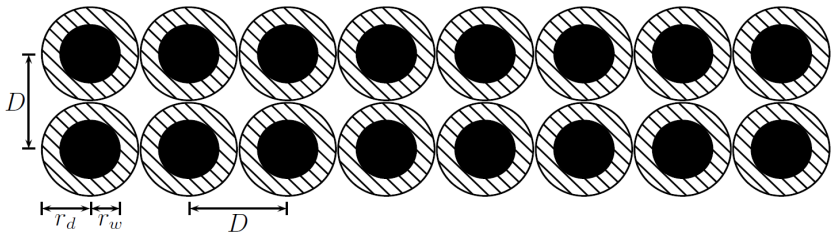


Figure A.6: Coupled Transmission Lines

The structure, presented also in [18, 19], has the following characteristics:

- wires length = 10cm;
- radius of the conductors $r_w = 0.5$ mm;
- radius of the dielectric insulator $r_d = 0.8$ mm;
- relative permittivity $\epsilon_r = 4.2$;
- distance between the wires center $D = 1.61$ mm.

Each conductors pair is considered as decoupled over a length $L - L_c$ while, for a length greater than L_c, all the conductors form a coupled $2N$ multi-conductor line. The free parameter is the coupling length L_c that is in the range [20, 40] mm. Parametric data have been extracted with a full-wave solver for 11 linearly spaced parameter values, each composed by a set of $\bar{k} = 500$ frequency samples.
Depending on the number of coupled wires N we consider, different models have been extracted.

Case 12

In this case we consider the structure with two differential pairs, thus $N = 2$. We set the partial fraction basis order $\bar{n} = 30$ and Chebyshev polynomial are used as parameter basis functions with cardinalities $\bar{\ell}_N = \bar{\ell}_D = 4$ for numerator and denominator, respectively. All the available datasets have been used to estimate the model.
The worst case relative error between raw fitting data and the PR-model is $9.1 \cdot 10^{-3}$, while for the not-PR model the error is $11.1 \cdot 10^{-3}$.

Case 13

Here the structure has three differential pairs, thus $N = 3$. The model orders have been set to $\bar{n} = 30$ for the partial fraction basis and to $\bar{\ell}_N = \bar{\ell}_D = 4$ for numerator and denominator parameter basis functions (Chebyshev polynomials). All the available datasets have been used to estimate the model.
The worst case relative error between raw fitting data and the PR-model is $13.0 \cdot 10^{-3}$, while for the not-PR model the error is $9.7 \cdot 10^{-3}$.

Case 14

Capacitor
This structure is a capacitor whose plates side-length, variable in [254, 609.6] μm, is assumed as free parameter. From repeated simulations we have 9 parametric datasets, each one composed of $\bar{k} = 191$ frequency samples, spanning the band [0.5, 10] GHz.
We estimated the model with partial fractions order $\bar{n} = 4$ and Chebyshev polynomial as parameter basis functions, with orders $\bar{\ell}_N = \bar{\ell}_D = 1$ for numerator and denominator, respectively. To extract the model we used 5 parametric datasets, leading the other 4 to be used for validation purposes.
The maximum relative error between the PR-model and validation data is $15.4 \cdot 10^{-3}$, while for the not-PR model the error is $13.1 \cdot 10^{-3}$.

Case 15

Two-Stage buffer
This structure is a two-stage buffer, depicted in Figure A.7, originally presented in [29]. This buffer is parameterized by the supply voltage V_{dd} and by the ambient temperature T, ranging, respectively, in [0.5, 1.5] V and [20, 40] °C.
The available dataset is composed of the frequency responses for 11 points along V_{dd} and 21 along T, linearly spaced in their ranges. For each parameters combination we have $\bar{k} = 293$ frequency samples, spanning the band [0, 100] THz. This frequency range is not indicative of the real component bandwidth, it has been exploited just for transistor-level simulations.
To estimate the model we used partial fractions order $\bar{n} = 6$ and Chebyshev polynomial as parameter basis function with orders $\bar{\ell}^1_N = 2$, $\bar{\ell}^1_D = 2$ for the first parameter (supply voltage) and $\bar{\ell}^2_N = 3$, $\bar{\ell}^2_D = 2$ for the second one (temperature). Among the available

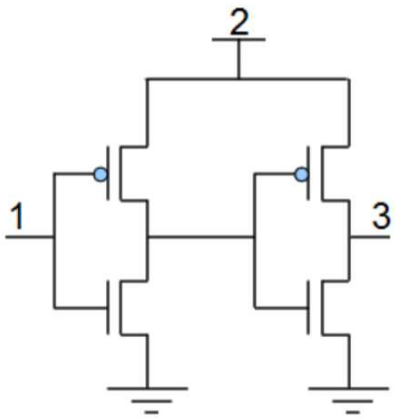


Figure A.7: Two-Stage Buffer

parametric datasets, we used half of them to estimate model coefficients and the others for validation purposes.
The maximum relative error between PR-model responses with respect to validation data is $28.4 \cdot 10^{-3}$, while for the not-PR model the error is $7.9 \cdot 10^{-3}$.

Case 16

Printed Circuit Board Interconnect
The structure underlying this test case is a high-speed PCB signal link (Courtesy of Prof. Christian Schuster and Dr. Jan Preibisch, Technische Universität Hamburg-Harburg, Hamburg, Germany), for details see [18, 22, 31, 47]. The signal path is a stripline routed in the inner layer of two PCB's, attached by a connector. The signal is provided to this connector by four vertical vias, whose pad and anti-pad radii are assumed as free parameters, varying respectively in [100, 300] μm and [400, 600] μm.
From repeated simulations, we have 81 frequency response datasets (9 for each parameter), each one containing $\bar{k} = 250$ frequency samples, spanning the band [1 Hz, 5 GHz]. We set partial fractions basis order to $\bar{n} = 24$ and parameters basis functions (Chebyshev polynomials) orders $\bar{\ell}_N^1 = 3$, $\bar{\ell}_D^1 = 2$ for the first parameter (pad radius) and $\bar{\ell}_N^2 = 3$, $\bar{\ell}_D^2 = 2$ for the second one (anti-pad radius). All the available datasets are used for fitting purposes.
The maximum relative error between raw fitting data and the PR-Model is $136 \cdot 10^{-3}$, while for the not-PR model is $14.3 \cdot 10^{-3}$.

Case 17

Inductor (Multi-parametric)
In this case, the structure is a 1-turn PCB plane inductor, with a square outline, placed on a dielectric substrate.
In details, we have a set of responses depending on two free parameters: the line width

and the dielectric thickness, variable in [76, 127] μm and [762, 1016] μm respectively. The dataset is composed of 9 sets of frequency responses (3 for each parameter), composed themselves of $\bar{k} = 699$ samples, spanning the band [100 MHz, 35 GHz]. We set as partial fractions basis order $\bar{n} = 8$ and Chebyshev polynomial as parameter basis functions with orders $\bar{\ell}_N^1 = 2$, $\bar{\ell}_D^1 = 2$ for the first parameter (line width) and $\bar{\ell}_N^2 = 1$, $\bar{\ell}_D^2 = 1$ for the second one (dielectric thickness). All the available data are used for fitting purposes.
The worst case relative error of the PR-model with respect to raw data is $14.3 \cdot 10^{-3}$, while for the not-PR one the error is $4.4 \cdot 10^{-3}$.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, 1968.

[2] B. Anderson and S. Vongpanitlerd. Network analysis and synthesis: A modern systems theory approach, eaglewood cli s, 1973.

[3] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[4] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the h norm of a transfer matrix and related problems. *Mathematics of Control, Signals and Systems*, 2(3):207–219, 1989.

[5] T. Bradde. Fast data-driven algorithms for parameterized macromodeling of multiport systems. Master's thesis, Politecnico di Torino, 2018.

[6] M. Bungarts, Hans-Joachim e Griebel. Sparse grids. *Acta numerica*, 13:147–269, 2004.

[7] T. Chihara. An introduction to orthogonal polynomials (gordon and breach science publishers, new york, ny). Technical report, ISBN 0-677-04150-0, 1978.

[8] M. De Stefano. Automated generation of stable bias-dependent small-signal behavioral macromodels for circuit-level simulation. Master's thesis, Politecnico di Torino, 2018.

[9] 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 *Microwave Conference (EuMC), 2018 European*. IEEE, 2018.

[10] D. Deschrijver and T. Dhaene. Parametric macromodeling of time domain responses. In *Signal Propagation on Interconnects, 2008. SPI 2008. 12th IEEE Workshop on*, pages 1–2. IEEE, 2008.

[11] F. Ferranti, L. Knockaert, and T. Dhaene. Parameterized s-parameter based macromodeling with guaranteed passivity. *IEEE Microwave and Wireless Components Letters*, 19(10):608–610, 2009.

[12] F. Ferranti, L. Knockaert, and T. Dhaene. Guaranteed passive parameterized admittance-based macromodeling. *IEEE Transactions on Advanced Packaging*, 33(3):623–629, 2010.

[13] F. Ferranti, L. Knockaert, and T. 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.

[14] T. Gerstner and M. Griebel. Sparse grids. *Encyclopedia of Quantitative Finance*, 2010.

[15] A. Gil, J. Segura, and N. M. Temme. *Numerical methods for special functions*,

volume 99. Siam, 2007.

[16] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins Univ Pr, 1996.

[17] S. Grivet-Talocia. Passivity enforcement via perturbation of hamiltonian matrices. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(9):1755–1769, 2004.

[18] S. 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.

[19] S. Grivet-Talocia, T. Bradde, M. De Stefano, and A. Zanco. A scalable reduced-order modeling algorithm for the construction of parameterized interconnect macromodels from scattering responses. In *IEEE Symposium on Electromagnetic Compatibility, Signal and Power Integrity*. IEEE, 2018.

[20] S. Grivet-Talocia and E. Fevola. Compact parameterized black-box modeling via fourier-rational approximations. *IEEE Transactions on Electromagnetic Compatibility*, 59(4):1133–1142, 2017.

[21] S. Grivet-Talocia and B. Gustavsen. *Passive macromodeling: Theory and applications*, volume 239. John Wiley & Sons, 2015.

[22] S. Grivet-Talocia and R. Trinchero. Behavioral, parameterized, and broadband modeling of wired interconnects with internal discontinuities. *IEEE Transactions on Electromagnetic Compatibility*, 60(1):77–85, 2018.

[23] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, Philadelphia, PA, 1996.

[24] T. Kailath. *Linear systems*. Prentice-Hall Englewood Cliffs, NJ, 1980.

[25] R. E. Kalman. On a new characterization of linear passive systems. In *First Allerton Conference on Circuit Theory*, 1963.

[26] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 1999.

[27] D. G. Luenberger. *Optimization by vector space methods*. Wiley-Interscience, 1997.

[28] A. F. Nikiforov, V. B. Uvarov, and R. P. Boas. *Special functions of mathematical physics*. Birkhäuser, 1988.

[29] S. B. Olivadese, G. Signorini, S. Grivet-Talocia, and P. 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.

[30] A. V. Oppenheim and A. S. Willsky. *Signals and systems*. Prentice Hall, Englewood Cliffs, NJ: Prentice Hall, 1983.

[31] J. 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.

[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] C. Sanathanan and J. Koerner. Transfer function synthesis as a ratio of two complex polynomials. *IEEE transactions on automatic control*, 8(1):56–58, 1963.

[34] C. Scherer and S. Weiland. Linear matrix inequalities in control. *Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands*, 3, 2000.

[35] P. Triverio, S. Grivet-Talocia, and M. Nakhla. An improved fitting algorithm for

parametric macromodeling from tabulated data. In *Signal Propagation on Interconnects, 2008. SPI 2008. 12th IEEE Workshop on*, pages 1–4. IEEE, 2008.

[36] P. Triverio, S. Grivet-Talocia, and M. S. Nakhla. A parameterized macromodeling strategy with uniform stability test. *IEEE Transactions on Advanced Packaging*, 32(1):205–215, 2009.

[37] P. Triverio, M. Nakhla, and S. Grivet-Talocia. Parametric macromodeling of multiport networks from tabulated data. In *Electrical Performance of Electronic Packaging, 2007 IEEE*, pages 51–54. IEEE, 2007.

[38] P. Triverio, M. Nakhla, and S. Grivet-Talocia. Extraction of parametric circuit models from scattering parameters of passive rf components. In *Microwave Conference (EuMC), 2010 European*, pages 1635–1638. IEEE, 2010.

[39] P. Triverio, M. Nakhla, and S. Grivet-Talocia. Passive parametric macromodeling from sampled frequency data. In *Signal Propagation on Interconnects (SPI), 2010 IEEE 14th Workshop on*, pages 117–120. IEEE, 2010.

[40] P. Triverio, M. S. Nakhla, and S. Grivet-Talocia. Parametric macromodeling of multiport networks from tabulated data. In *IEEE* 16^{th} *Topical Meeting on Electrical Performance of Electronic Packaging (EPEP 2007), Atlanta, GA*, pages 51–54, October 29–31, 2007.

[41] C. F. Van Loan. Matrix computations (johns hopkins studies in mathematical sciences), 1996.

[42] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer. Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Processing Magazine*, 31(5):71–79, 2014.

[43] Y. Wang, Z. Zhang, C.-K. Koh, G. Shi, G. K. Pang, and N. Wong. Passivity enforcement for descriptor systems via matrix pencil perturbation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(4):532–545, 2012.

[44] M. Wohlers and E. Beltrami. Distribution theory as the basis of generalized passive-network analysis. *IEEE Transactions on Circuit Theory*, 12(2):164–170, 1965.

[45] M. R. Wohlers. *Lumped and Distributed Passive Networks*. Academic press, 1969.

[46] D. Youla, L. Castriota, and H. Carlin. Bounded real scattering matrices and the foundations of linear passive network theory. *IRE Transactions on Circuit Theory*, 6(1):102–124, 1959.

[47] A. Zanco, S. Grivet-Talocia, T. Bradde, and M. De Stefano. Multivariate macromodeling with stability and passivity constraints. In *Signal Propagation on Interconnects, 2018. SPI 2018. 22th IEEE Workshop on*. IEEE, 2018.

[48] Z. Zhang, K. Batselier, H. Liu, L. Daniel, and N. Wong. Tensor computation: A new framework for high-dimensional problems in eda. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(4):521–536, 2017.