Fast macromodeling of large-scale multiports with guaranteed stability

Fast macromodeling of large-scale multiports with guaranteed stability

1st Tommaso Bradde
Department of Electronics and Telecommunications
Politecnico di Torino
Turin, Italy
tommaso.bradde@polito.it

2nd Ion Victor Gosea
Max Planck Institute
for Dynamics of Complex Technical Systems
Magdeburg, Germany
gosea@mpi-magdeburg.mpg.de

3rd Stefano Grivet-Talocia
Department of Electronics and Telecommunications
Politecnico di Torino
Turin, Italy
stefano.grivet@polito.it

***Abstract*—This contribution introduces a novel approach for generating guaranteed stable macromodels of large multiport structures in a completely automated and efficient manner. The presented method is based on the Adaptive Antoulas-Anderson (AAA) algorithm for rational fitting of scalar transfer functions. We propose a computationally cheap multi-input multi-output extension of the AAA, and we combine the resulting algorithm with a novel post-processing stability enforcement step that is formulated in terms of a small-size convex program. Applying the resulting framework to a large Power Delivery Network (PDN), we show a significant computational cost reduction with respect to commonly employed state-of-the-art methods. The proposed scheme fits naturally as a bridge between electromagnetic and circuit simulation, enabling the representation of high-frequency phenomena and parasitics as low-order equivalent circuits synthesized from the computed macromodels.**

***Index Terms*—Macromodeling, power integrity, transient analyses.**

I. Introduction

Transient analysis of complex electronic systems represents nowadays a mandatory step for the verification and the optimization of the electrical performance of a given design, especially for meeting adequate signal and power integrity requirements. To cope with the computational complexity that characterizes such simulations, rational macromodels of large-scale passive components are of crucial importance. Beyond being in general accurate and compact, macromodels allow for conversion between native characterizations of the underlying components given in terms of frequency domain tabulated responses into small size equivalent circuits, thus enabling the possibility of performing fast transient analysis when only frequency data are available. Nowadays, the Vector Fitting (VF) iteration [1] is the method of choice for generating accurate macromodels in a data-driven setting, mainly due to its robustness [2] and ease in enforcing model stability.

Despite the algorithm admits efficient [3] and parallelization compliant implementations [4], the continuously increasing complexity of state-of-the-art electronic technologies often requires handling extremely large scale structures, for which the synthesis of the macromodel represents by itself a time-demanding task. This is particularly true when dealing with system-level power integrity verification of Power Delivery Networks (PDNs), that are defined in terms of hundreds or even thousands of electrical ports [5] and pose significant challenges to the computational feasibility of the macromodeling process. To boost the efficiency of the macromodeling workflow in the presence of a large number of electrical ports, the works in [6], [7] propose to exploit data redundancy by reducing the rational fitting problem to that of reconstructing via VF only a limited number of independent principal components, from which the desired response can be recovered by superposition.

In this work, we introduce a novel approach for modeling large-scale electrical interconnects in an efficient and automated manner, that does not rely on any data redundancy assumption or need for potentially costly preprocessing. Contrarily to the available state-of-the-art approaches, the proposed method concurrently retains the following advantages

1) is based on a model structure with a number of free variables that is independent of the number of underlying structure ports,
2) simultaneously performs model generation and order estimation, requiring as input only a user-prescribed accuracy requirement, and
3) guarantees asymptotically stable models by construction.

Our approach represents a Multi-Input-Multi-Output (MIMO) formulation of the recently introduced Adaptive Antoulas-Anderson (AAA) algorithm for scalar rational approximation [8] and performs an iterative update of a template barycentric model structure until it meets a prescribed target accuracy. The model update is performed by first interpolating the model response against the data point where the maximum residual

error is detected, and then optimizing a set of free model coefficients to approximate the remaining samples. Due to the interpolation conditions, the number of such model coefficients depends only on the current iteration number. Further, the algorithm admits a particularly efficient implementation that does not require computing from scratch large data matrices as the model complexity increases.

From the applications standpoint, one of the main reasons that prevented interpolatory approaches from being successfully exploited in macromodeling workflows was the lack of stability guarantees, that are instead easily provided by approximation schemes similar to VF. In this sense, we propose for the first time a novel set of algebraic conditions that can be used to characterize and enforce the stability of the employed model structure without spoiling its accuracy. The stability enforcement is achieved via post-processing, and requires the solution of a small semidefinite program, easily handled via convex optimization methods in a completely deterministic and efficient manner.

Experimental evidence based on macromodeling of a realistic PDN structure involving hundreds of electrical ports shows that the proposed approach is superior to the VF iteration in terms of efficiency, granting a reduction of the modeling time requirements of almost $50\times$ while being on par in terms of modeling accuracy. Thus, it represents a valuable tool for increasing the efficiency of power integrity assessment workflows.

II. Notation and Problem statement

We will denote with $j = \sqrt{-1}$ and with $s = \sigma + j\omega \in \mathbb{C}$ respectively the imaginary unit and the Laplace variable. Lowercase italic letters will be reserved for scalars, while lowercase and uppercase blackboard bold letters will be reserved for vectors and matrices (e.g., $\mathbf{x}$ for a vector and $\mathbf{X}$ for a matrix). Given a matrix $\mathbf{X}$, its (r, c)-th element will be denoted as x_{rc}. With $\mathbf{X}^T$ we will denote the matrix transpose. The matrix $\mathbf{I}_n$ is the identity matrix of size n. The expression $\mathbf{J} \succ \mathbf{K}$ ($\mathbf{J} \succeq \mathbf{K}$) means that the matrix $\mathbf{J}-\mathbf{K}$ is positive (semi)definite. Given a matrix, we will refer to least eigen/singular vector as the one associated with its smallest eigen/singular value.

We consider an arbitrarily complex electromagnetic structure accessible from the outer environment by a large set of P well-defined electrical ports. We assume that the knowledge of this structure is limited to the availability of samples of one of its network functions, e.g. its impedance, admittance or scattering matrix; we will generically denote this network function in the following as $\mathbf{H}(s) \in \mathbb{C}^{P\times P}$. Further, we assume that the available data represent samples of $\mathbf{H}(s)$ retrieved in correspondence of a set Γ of discrete frequency configurations,

$$\mathbf{H}_v = \mathbf{H}(j\lambda_v), \quad j\lambda_v \in \Gamma = \{j\lambda_1, \ldots, j\lambda_V\},\ \lambda_v > 0. \quad (1)$$

Such samples can be obtained through real or virtual measurements, performed in the latter case exploiting a high-fidelity electromagnetic (full-wave) characterization of the structure, including high-performance commercial Maxwell equations solvers.

In this setting, our goal is to formulate an efficient numerical algorithm that returns an asymptotically stable macromodel with rational transfer function $\hat{\mathbf{H}}(s)$ fulfilling the following approximation condition

$$\hat{\mathbf{H}}(j\lambda_v) \approx \mathbf{H}_v, \quad \forall j\lambda_v \in \Gamma. \quad (2)$$

III. Model structure

Our approach builds the macromodel network function $\hat{\mathbf{H}}(s)$ based on the following formulation

$$\hat{\mathbf{H}}(s) = \frac{\sum_{i=1}^{k}\left(\frac{\mathbf{F}^i w_i}{s-j\lambda_i} + \frac{(\mathbf{F}^i w_i)^*}{s+j\lambda_i}\right)}{\sum_{i=1}^{k}\left(\frac{w_i}{s-j\lambda_i} + \frac{w_i^*}{s+j\lambda_i}\right)} = \frac{\mathbf{N}(s)}{d(s)} \in \mathbb{C}^{P\times P}, \quad (3)$$

that represents a straightforward matrix extension [9]–[12] of the barycentric structure employed in many state-of-the-art rational fitting approaches, including the scalar AAA algorithm [8], [13], and the classical Loewner matrix interpolation framework [14]. See also [15], [16], [16]–[19]. Other advances of the AAA algorithm, including approximation on a continuum and randomized sketching properties, were shown in [20], [21].

In (3), the barycentric weights $w_i = \alpha_i + j\beta_i \in \mathbb{C}$ represent the model unknowns that must be optimized in order to enforce condition (2), while the matrices $\mathbf{F}^i \in \mathbb{C}^{P\times P}$ are assumed to be known quantities. Whenever $w_i \neq 0, i = 1 \ldots k$ this model structure retains the following properties by construction

- Every element of $\hat{\mathbf{H}}(s)$ can be expressed as a pole-residue expansion over the same set of at most $2k-1$ poles. Depending on w_i, these poles can be located everywhere except at locations $\pm j\lambda_i$ [8].
- It holds $\hat{\mathbf{H}}^*(s) = \hat{\mathbf{H}}(s^*)$ so that the impulse response of the macromodel is real-valued.
- It holds $\hat{\mathbf{H}}(j\lambda_i) = \mathbf{F}^i$. The quantities $\pm j\lambda_i$ are referred to as support points [8].

Due to the last interpolation property, by selecting $\lambda_i \in \Gamma$ and $\mathbf{F}^i = \mathbf{H}(j\lambda_i)$ the macromodel will interpolate the underlying structure response over a subset of the available sample points. The next section outlines an algorithm derived from AAA [8] that efficiently and automatically selects the interpolation points (and thus the model order) and estimates the model coefficients in order to meet the approximation condition (2).

IV. Model generation

Following the philosophy of [8], the proposed approach builds a model of the kind of (3) following a greedy iterative procedure. By defining the iteration index ℓ, and denoting with $\hat{\mathbf{H}}_\ell(s)$ the model obtained at the ℓ-th iteration, the algorithm is initialized by defining for $\ell = 0$

$$\hat{\mathbf{H}}_0(s) = V^{-1}\sum_{v=1}^{V}\mathbf{H}_v. \qquad \Gamma^{(0)} = \Gamma. \quad (4)$$

The ℓ-th iteration ($\ell \geq 1$) starts by finding the data point $j\lambda_\ell$ within $\Gamma^{(\ell-1)}$ over which the maximum error value is attained, according to the following criterion

$$j\lambda_\ell = \arg \max_{j\lambda \in \Gamma^{(\ell-1)}} \left\| \mathbf{H}(j\lambda) - \hat{\mathbf{H}}_{\ell-1}(j\lambda) \right\|_2 \tag{5}$$

and consequently updating the definition of the test set $\Gamma^{(\ell)} := \Gamma^{(\ell-1)} \setminus \{j\lambda_\ell\}$ and of the model as

$$\hat{\mathbf{H}}_\ell(s) := \frac{\sum_{i=1}^{\ell} \left(\frac{\mathbf{H}^i w_i^{(\ell)}}{s-j\lambda_i} + \frac{(\mathbf{H}^i w_i^{(\ell)})^*}{s+j\lambda_i} \right)}{\sum_{i=1}^{\ell} \left(\frac{w_i^{(\ell)}}{s-j\lambda_i} + \frac{(w_i^{(\ell)})^*}{s+j\lambda_i} \right)} = \frac{\mathbf{N}_\ell(s)}{d_\ell(s)}, \tag{6}$$

with $\mathbf{H}^\ell = \mathbf{H}(j\lambda_\ell)$. The model unknowns $w_i^{(\ell)}$ are found in order to enforce (2) using the current model structure. This is done by defining the linearized error matrix function,

$$\mathbf{E}_\ell(s) = d_\ell(s)\mathbf{H}(s) - \mathbf{N}_\ell(s), \tag{7}$$

with (r, c)-th entries that read

$$e_{rc}^\ell(s) = \sum_{i=1}^{\ell} \frac{h_{rc}(s) - h_{rc}^i}{s - j\lambda_i} \cdot w_i^{(\ell)} + \sum_{i=1}^{\ell} \frac{h_{rc}(s) - (h_{rc}^i)^*}{s + j\lambda_i} \cdot (w_i^{(\ell)})^*, \tag{8}$$

and looking for the $w_i^{(\ell)}$ that best meet the condition

$$\Re\{e_{rc}^\ell(s)\} \approx 0,\ \Im\{e_{r,c}^\ell(s)\} \approx 0 \quad \forall s \in \Gamma^{(\ell)}, \quad r, c = 1, \dots P. \tag{9}$$

Since the functions $e_{rc}^\ell(s)$ are linear in the unknowns $w_i^{(\ell)} = \alpha_i^{(\ell)} + j\beta_i^{(\ell)}$, the ensemble of conditions (9) can be expressed collectively in matrix form involving only real quantities, as for similar rational fitting schemes [1], [2]. Defining the vector of real unknowns $\mathbf{x}^{(\ell)} = [\alpha_1^{(\ell)}, \beta_1^{(\ell)}, \dots, \alpha_\ell^{(\ell)}, \beta_\ell^{(\ell)}]^T$, conditions (9) are rewritten as

$$\mathbf{L}^{(\ell)}\mathbf{x}^{(\ell)} \approx 0, \quad \mathbf{L}^{(\ell)} \in \mathbb{R}^{2(V-\ell)P^2 \times 2\ell}. \tag{10}$$

The above error minimization condition is enforced in a least-squares sense, solving the homogeneous problem

$$\mathbf{x}_{\text{opt}}^{(\ell)} = \arg \min_{||\mathbf{x}^{(\ell)}||_2 = 1} \left\| \mathbf{L}^{(\ell)}\mathbf{x}^{(\ell)} \right\|_2. \tag{11}$$

The solution to the above optimization problem is the least right singular vector of $\mathbf{L}^{(\ell)}$. Once the solution is available, the optimal model coefficients are used to set up the next iteration.

The iteration stops when the fitting error hits a user-prescribed error tolerance. As target error metric, in this work, we consider the following

$$\delta^{(\ell)} = \sqrt{\frac{1}{VP^2} \sum_{r=1}^{P} \sum_{c=1}^{P} \sum_{v=1}^{V} \left(h_{rc}(j\lambda_v) - \hat{h}_{rc}^{(\ell)}(j\lambda_v) \right)^2}, \tag{12}$$

that represents the cumulative RMS error of the model with respect to the training dataset. Using this definition, the algorithm stop condition is $\delta^{(\ell)} \leq \epsilon$, being ϵ a positive user-defined error tolerance. The following remarks are in order

- The support point selection criterion (5) is chosen as a generalization of the criterion used in [8] for the scalar case. The algorithm selects as the next support point the frequency sample for which the largest singular value of the residual error matrix attains its maximum.
- To stop the algorithm, we use the RMS value $\delta^{(\ell)}$ defined in (12), and not the same error metric that drives the support point selection. This is because the RMS error is insensitive to the dimension P of the underlying structure and representative of the overall model accuracy. We highlight however that different error metrics can be used to define the stop criterion.

V. Efficient Implementation

From the computational standpoint, each iteration of the algorithm described in Sec. IV amounts to evaluating the samples of the cost function in (5), constructing the regression matrix $\mathbf{L}^{(\ell)}$, and computing its SVD. The latter two steps admit a particularly efficient implementation. This can be seen by noticing that at iteration ℓ one as

$$\mathbf{L}^{(\ell)} = \begin{bmatrix} \mathbf{B}^{(\ell)} & \mathbf{C}^{(\ell)} \end{bmatrix}, \quad \mathbf{C}^{(\ell)} \in \mathbb{R}^{2(V-\ell)P^2 \times 2}, \tag{13}$$

where $\mathbf{B}^{(\ell)}$ is obtained from $\mathbf{L}^{(\ell-1)}$ by removing the rows corresponding to the evaluation of the error functions $\Re\{e_{rc}^{(\ell-1)}(s)\}, \Im\{e_{rc}^{(\ell-1)}(s)\}$ over the newly defined support point $j\lambda_\ell$, while $\mathbf{C}^{(\ell)}$ is the actual update term needed to take into account the additional variable $w_\ell = \alpha_\ell + j\beta_\ell$ introduced in (8) by the current iteration. Therefore, the matrix $\mathbf{L}^{(\ell)}$ must not be recomputed from scratch when a new support point is added. Also, we recall that the least right singular vector of $\mathbf{L}^{(\ell)}$ computed to solve (11) coincides with the least eigenvector of the symmetric matrix $\mathbf{R}^{(\ell)} = \left(\mathbf{L}^{(\ell)}\right)^T \mathbf{L}^{(\ell)} \in \mathbb{R}^{2\ell \times 2\ell}$. In view of (13), this matrix can be partitioned as

$$\mathbf{R}^{(\ell)} = \begin{bmatrix} \left(\mathbf{B}^{(\ell)}\right)^T \mathbf{B}^{(\ell)} & \left(\mathbf{B}^{(\ell)}\right)^T \mathbf{C}^{(\ell)} \\ \left(\mathbf{C}^{(\ell)}\right)^T \mathbf{B}^{(\ell)} & \left(\mathbf{C}^{(\ell)}\right)^T \mathbf{C}^{(\ell)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{11}^{(\ell)} & \mathbf{R}_{12}^{(\ell)} \\ \mathbf{R}_{21}^{(\ell)} & \mathbf{R}_{22}^{(\ell)} \end{bmatrix}, \tag{14}$$

and can be built efficiently by

- Computing from scratch $\mathbf{R}_{22}^{(\ell)}$ and $\mathbf{R}_{12}^{(\ell)} = \left(\mathbf{R}_{21}^{(\ell)}\right)^T$
- Computing $\mathbf{R}_{11}^{(\ell)}$ from $\mathbf{R}^{(\ell-1)}$ via the update

$$\mathbf{R}_{11}^{(\ell)} = \mathbf{R}^{(\ell-1)} - \left(\mathbf{\Lambda}^{(\ell)}\right)^T \mathbf{\Lambda}^{(\ell)} \tag{15}$$

where $\mathbf{\Lambda}^{(\ell)}$ stacks the rows removed from $\mathbf{L}^{(\ell-1)}$ to obtain $\mathbf{B}^{(\ell)}$.

Once $\mathbf{R}^{(\ell)}$ is available, computing its eigendecomposition is a computationally inexpensive task since it is square symmetric with dimensions equal to the current model complexity (i.e. order, which is assumed to be small). Neglecting the cost of this last operation, and supposing that the algorithm stops at iteration k, the cumulative computational cost required to optimize the model coefficients during all the iterations is approximately the same as building $\mathbf{L}^{(k)}$ from scratch and computing the product $\left(\mathbf{L}^{(k)}\right)^T \mathbf{L}^{(k)}$.

VI. Stability Enforcement

Let us consider the model structure in (3). Due to its barycentric form, the poles of each element of $\hat{\mathbf{H}}(s)$ are located in correspondence with the zeros of the denominator $d(s)$. Applying the model generation routine outlined in Sec. IV does not guarantee that such zeros have a strictly negative real part, meaning that the procedure could generate an unstable model. Thus, the admissible model coefficients w_i should be restricted to the set of those such that $d(s)$ is a *minimum phase function* [22], i.e., a function whose zeros are located in the open left-half complex plane. To introduce our stability enforcement approach, we will need the following definitions

Definition VI.1 (Positive Real (PR) function [23])**.** *A scalar rational function $f(s)$ is positive real if*

1) *$f(s)$ has no poles in $\Re\{s\} > 0$*
2) *$f(s)$ is real for all positive real s*
3) *$\Re\{f(s)\} \geq 0$ for $\Re\{s\} > 0$*

Definition VI.2 (Strictly Positive Real (SPR) function [23])**.** *A rational function $f(s) \in \mathbb{C}$ that is not identically zero for all s is strictly positive real if $f(s-\tau)$ is PR for some $\tau > 0$.*

The above definitions are of interest because of the following result, which is proved in [24] for the case $P = 1$, and extended here for the multiport case:

Lemma VI.1. *Let $\hat{\boldsymbol{H}}(s)$ be defined as in* (3)*, with $w_i \neq 0$. Assume that all the elements of $\hat{\boldsymbol{H}}(s)$ have $2k-1$ poles and that $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ is a minimal state space realization for $d(s)$, with $\boldsymbol{cb} > 0$. Then $\hat{\boldsymbol{H}}(s)$ is asymptotically stable if and only if $\exists g \in \mathbb{R}$ such that*

$$g^+(s) = \frac{d(s)}{1 + g d(s)} = \boldsymbol{c}(s\boldsymbol{I}_{2k} - \boldsymbol{A} + g\boldsymbol{bc})^{-1}\boldsymbol{b} \qquad (16)$$

is SPR.

Building on the proof reported in [24] for the case $P = 1$ and based on the employed model structure, the validity of this result for an arbitrary number of ports P can be verified easily. About the hypothesis of the Lemma, we remark the following:

- From the numerical standpoint, the condition $w_i \neq 0$ is always verified in practice.
- The assumption that $d(s)$ admits a realization with $\mathbf{cb} > 0$ is not restrictive, as will be clear in the following.
- When model structure (3) is defined upon k support points, $w_i \neq 0$, and $\mathbf{cb} > 0$, all the elements of $\hat{\mathbf{H}}(s)$ have $2k-1$ poles up to exact zero-poles cancellations, that practically never occur numerically.

Based on Lemma VI.1, the stability of $\hat{\mathbf{H}}(s)$ can be enforced as follows. Let us assume that the modeling procedure of Sec. IV stops at iteration k. Dropping any iteration index, we consider the case in which the final model, defined by the set of optimal coefficients $\mathbf{x}_{\text{opt}}$, obtained by solving (11), is unstable. Our objective is to estimate a new set of weights $w_i = \alpha_i + j\beta_i, i = 1, \ldots, k$ that renders the macromodel stable while preserving its accuracy. The denominator $d(s)$ of the target stable model can be represented as

$$d(s) = \sum_{i=1}^{k} \left(\frac{w_i}{s - j\lambda_i} + \frac{w_i^*}{s + j\lambda_i} \right) = \mathbf{c}(s\mathbf{I}_{2k} - \mathbf{A})^{-1}\mathbf{b}, \quad (17)$$

with state space realization

$$\mathbf{A} = \text{blkdiag}[\mathbf{A}_1, \ldots, \mathbf{A}_k], \qquad \mathbf{b} = [\mathbf{b}_1, \ldots, \mathbf{b}_k]^T, \qquad (18)$$

$$\mathbf{c} = [\alpha_1, \beta_1, \ldots, \alpha_k, \beta_k], \qquad (19)$$

and

$$\mathbf{A}_i = \begin{bmatrix} 0 & \lambda_i \\ -\lambda_i & 0 \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} 2 & 0 \end{bmatrix}, \quad i = 1, \ldots, k. \qquad (20)$$

The definition (19) shows that $\mathbf{c}$ embeds the model unknowns in a vector defined as the unknown vector in condition (10). In terms of such unknowns, the fitting problem (11) can be rewritten equivalently exploiting $\mathbf{x}_{\text{opt}}$ as

$$\mathbf{c}_{\text{opt}} = \arg\min_{\mathbf{c}} \left\| \mathbf{L}(\mathbf{c}^T - \mathbf{x}_{\text{opt}}) \right\|_2^2, \qquad (21)$$

which is not homogeneous, and where we assume that the ambiguity on the sign of the singular vector $\mathbf{x}_{\text{opt}}$ is removed by requiring $\mathbf{x}_{\text{opt}}^T\mathbf{b} > 0$. This choice is not restrictive, since changing the sign of $\mathbf{x}_{\text{opt}}$ (or, equivalently, of $\mathbf{c}_{\text{opt}}$) has no influence on model structure (3).

To enforce stability, we will consider as feasible for problem (21) only the vectors $\mathbf{c}$ that render SPR the function $g^+(s)$ in (16) for some $g \in \mathbb{R}$. Under the hypothesis of Lemma VI.1, this condition is equivalent to the stability of $\hat{\mathbf{H}}(s)$. In view of the Positive Real Lemma [25], the SPR-ness of $g^+(s)$ can be verified by every feasible solution of the following constrained optimization problem

$$\begin{aligned} \mathbf{c}_{\text{opt}} = \arg\min_{\mathbf{c},\mathbf{Q},g} & \left\| \mathbf{L}(\mathbf{c}^T - \mathbf{x}_{\text{opt}}) \right\|_2^2, \qquad (22) \\ & \text{subject to:} \\ & \mathbf{Q} \in \mathbb{R}^{2k \times 2k},\ \mathbf{Q} = \mathbf{Q}^T \succ 0, \\ & \mathbf{A}^T\mathbf{Q} + \mathbf{Q}\mathbf{A} - g(\mathbf{bc})^T\mathbf{Q} - g\mathbf{Q}\mathbf{bc} \prec 0, \\ & \mathbf{Q}\mathbf{b} = \mathbf{c}^T. \end{aligned}$$

Verification steps for this condition are omitted here for the sake of brevity, but they can be performed as in [24]. Notice that $\mathbf{Qb} = \mathbf{c}^T$ and $\mathbf{Q} \succ 0$ enforced together imply $\mathbf{cb} > 0$, as required by Lemma VI.1. Thus, solving (22) returns a vector of model weights $\mathbf{c}_{\text{opt}}$ that guarantees the stability of $\hat{\mathbf{H}}(s)$ by construction.

Unfortunately, problem (22) is non-convex and hard to solve even when a small number of variables is involved. To tackle the problem, here we follow the approach introduced in [24], which relies on solving a convex relaxation of (22), formulated in terms of a semi-definite program involving $(4k^2 + 2)$ variables. Full details about the relaxation approach are available in [24, Sec. 4.2, 4.3]. For the sake of our discussion, we highlight here that the employed approach preserves the exactness of the stability constraints, and that its complexity depends only on the number of poles of the macromodel (that

is assumed to be small). Thus, the computational requirements of the proposed stability enforcement step are independent on the number of ports P of the underlying structure.

VII. Experimental Results

We apply the proposed approach to model a real PDN design employed on a 4-core mobile computational platform (courtesy of Intel) with $P = 144$ electrical ports, first presented in [5]. The structure is known through a high-accuracy mixed circuit-electromagnetic characterization given in terms of samples of its impedance matrix $\mathbf{Z}(s)$. The total number of sampling points for this test case is $V = 84$, logarithmically distributed in the interval $f \in [10^{-8}, 2.5]$ GHz.

The algorithm is applied with RMS error tolerance $\epsilon = 5 \times 10^{-6}$. With this setting, the model stops at the 13-rd iteration, returning a rational macromodel of order 25. The RMS error committed by the model is $\delta^{(13)} = 2.2348 \times 10^{-6}$; yet, it exhibits one unstable real pole, so that the stability enforcement step described in Sec. VI must be applied. After the model correction, the asymptotically stable result commits the RMS error $\delta^{(13)}_{stab} = 2.2352 \times 10^{-6}$. For this test case, the stability enforcement practically does not introduce any significant accuracy degradation.

A graphical illustration of the model fitting accuracy is reported in Figs. 1 and 2, which respectively show comparisons between data and model responses for diagonal and off-diagonal entries of the PDN impedance matrix. An overall picture of the RMS residual error is provided in Fig. 3; such error is bounded by 2.0511×10^{-5} over the $P \times P$ responses.

Our MATLAB implementation was executed on a workstation equipped with 32 GB of memory and a 3.3 GHz Intel i9-X7900 CPU. The stability enforcement step was performed using the YALMIP toolbox [26] and the semidefinite programming solver MOSEK [27]. The whole modeling process took 12.6 s, where $\approx$ 10.14 s were necessary to build the model while $\approx$ 1.6 s were spent in stability enforcement.

To further assess the scalability features of the workflow with respect to the number of underlying structure ports, we used the available data to simulate the scenario in which a PDN with $P = 288$ ports is considered. This was done by defining a fictitious impedance matrix

$$\mathbf{Z}_{ext}(s) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \mathbf{Z}(s) \in \mathbb{C}^{288 \times 288}, \tag{23}$$

where $\otimes$ is the Kronecker product. Defining a new dataset accordingly, we modeled the resulting synthetic structure using our algorithm. The runtime required in this case amounted to $\approx$ 44 s, of which 1.5 s were taken by stability enforcement and the remaining by the model generation. As expected by the structure of $\mathbf{Z}_{ext}(s)$, the quality of the model is in perfect agreement with the one obtained when modeling $\mathbf{Z}(s)$.

As a term of comparison, we modeled both $\mathbf{Z}(s)$ and $\mathbf{Z}_{ext}(s)$, using the same dataset, via Fast Vector Fitting (FVF) [3], setting the algorithm to return a model of order 25 after 10 iterations. For the smaller test case, FVF took $\approx$ 119 s, while for the larger one, it took $\approx$ 2050 s. Thus, the proposed approach guarantees respectively a speed-up factor of $9.41\times$ and $46.61\times$ when compared with this state-of-the-art method. Higher gains are expected for increasing port count. The FVF iteration fitted the samples of $\mathbf{Z}(s)$ with a residual error $\delta_{VF} = 1.0084 \times 10^{-6}$, which is fully compatible with the accuracy of our proposed method so that the two models can be considered as equivalent up to any practical extent.

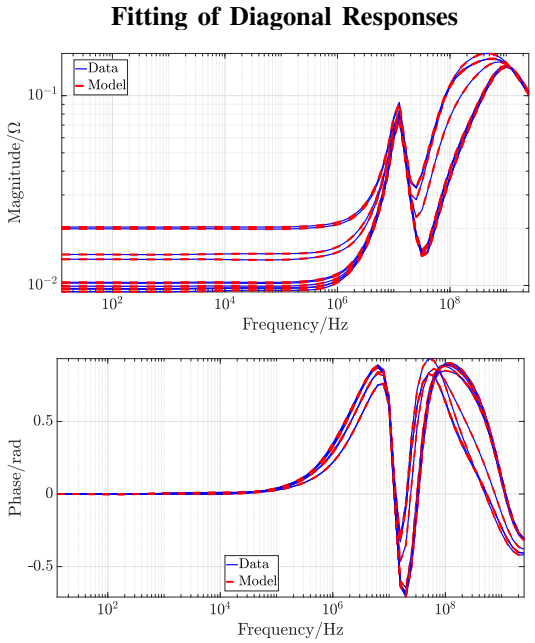

Fig. 1. Modeling performance of the proposed approach on a PDN structure with $P = 144$ electrical ports. The figure shows the quality of the model fitting against the available data over a number of diagonal elements of the PDN impedance matrix.

We note that the above results were obtained by executing prototypal MATLAB implementations of both the proposed approach and the FVF. We also tested the former in terms of efficiency against a state-of-the-art implementation of FVF, by modeling $\mathbf{Z}_{ext}(s)$ using a commercial modeling tool [28]. The tool took $\approx$ 95 s to process the dataset, returning a macromodel with a residual error which is practically identical to our FVF MATLAB implementation. Even when compared with this commercial tool, the proposed approach produces a stable model in less than half of the run time. This suggests additional margins for improvement in speedup, which can be attained by code optimization of the proposed algorithm.

VIII. Conclusions

In this contribution, we introduced a novel macromodeling framework specifically designed to efficiently handle large-scale electrical interconnects, such as the PDNs that are commonly encountered when dealing with transient analyses for power integrity. The approach represents a valuable tool for reducing the time required by the overall electrical performance assessment workflow, by providing users with a reliable, efficient, and fully automated algorithm for performing system-level verification and optimization. The method can be easily embedded in any EDA tool or flow to convert

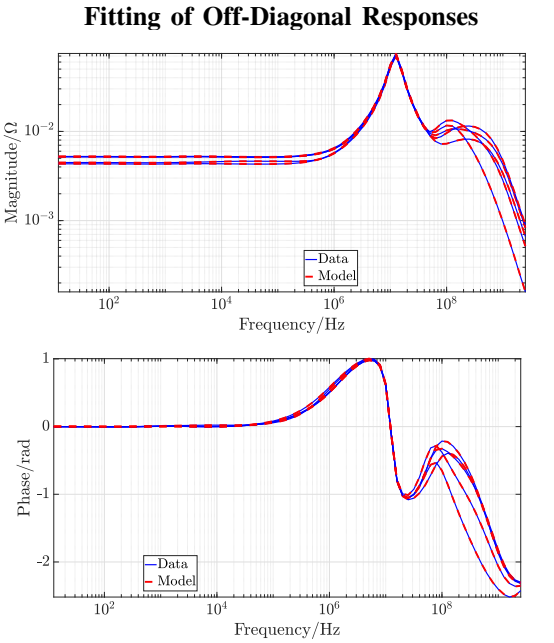

Fig. 2. As in Fig. 1, but for off-diagonal elements of the impedance matrix.

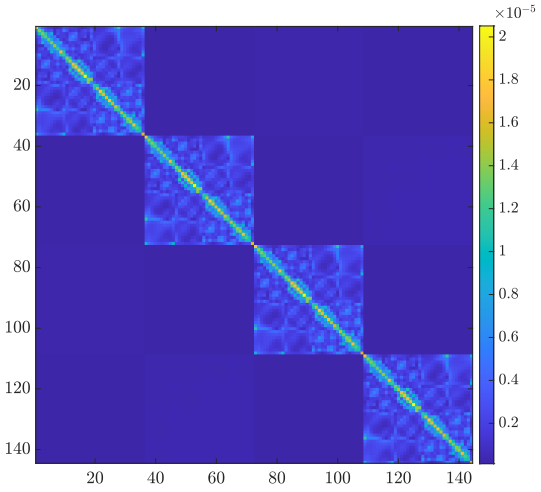

Fig. 3. A graphical representation of the RMS error committed by the macromodel against each element of the PDN impedance matrix.

large-scale S-parameter sets to SPICE-compatible equivalent circuits, enabling significant modeling time reductions when compared to state-of-the-art approaches. Further investigations will concern the efficient passivity enforcement of massively MIMO systems, such as those considered in this paper.

References

[1] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1052–1061, 1999.

[2] B. Gustavsen, "Improving the pole relocating properties of vector fitting," *IEEE Trans. Power Del.*, vol. 21, no. 3, pp. 1587–1592, 2006.

[3] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *Microwave and Wireless Components Letters, IEEE*, vol. 18, no. 6, pp. 383–385, june 2008.

[4] A. Chinea and S. Grivet-Talocia, "On the parallelization of vector fitting algorithms," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 11, pp. 1761–1773, 2011.

[5] A. Carlucci, T. Bradde, S. Grivet-Talocia, S. Mongrain, S. Kulasekaran, and K. Radhakrishnan, "A compressed multivariate macromodeling framework for fast transient verification of system-level power delivery networks," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 13, no. 10, pp. 1553–1566, 2023.

[6] S. B. Olivadese and S. Grivet-Talocia, "Compressed passive macromodeling," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 2, no. 8, pp. 1378–1388, 2012.

[7] M. De Stefano, T. Wendt, C. Yang, S. Grivet-Talocia, and C. Schuster, "Regularized and compressed large-scale rational macromodeling: Theory and application to energy-selective shielding enclosures," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 5, pp. 1365–1379, 2022.

[8] Y. Nakatsukasa, O. Sete, and L. N. Trefethen, "The AAA algorithm for rational approximation," *SIAM J. Sci. Comput.*, vol. 40, no. 3, pp. A1494–A1522, 2018.

[9] I. V. Gosea and S. Güttel, "Algorithms for the rational approximation of matrix-valued functions," *SIAM J. Sci. Comput.*, vol. 43, no. 5, pp. A3033–A3054, 2021.

[10] I. V. Gosea, C. Poussot-Vassal, and A. C. Antoulas, "On enforcing stability for data-driven reduced-order models," in *2021 29th Mediterranean Conference on Control and Automation (MED)*, 2021, pp. 487–493.

[11] L. Monzón, W. Johns, S. Iyengar, M. Reynolds, J. Maack, and K. Prabakar, "A multi-function aaa algorithm applied to frequency dependent line modeling," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[12] L. Davis, W. Johns, L. Monzón, and M. Reynolds, "Iterative stability enforcement in adaptive antoulas–anderson algorithms for model reduction," *SIAM J. Sci. Comput.*, vol. 45, no. 4, pp. A1844–A1861, 2023.

[13] A. Valera-Rivera and A. E. Engin, "AAA algorithm for rational transfer function approximation with stable poles," *IEEE Lett. on Electromagn. Compat. Prac. and App.*, vol. 3, no. 3, pp. 92–95, 2021.

[14] A. C. Antoulas and B. D. O. Anderson, "On the scalar rational interpolation problem," *IMA J. Math. Control. Inf.*, vol. 3, no. 2-3, pp. 61–88, 1986.

[15] Q. Aumann and I. V. Gosea, "Practical challenges in data-driven interpolation: dealing with noise, enforcing stability, and computing realizations," *Internat. Jour. of Adapt. Contr. and Signal Proc.*, 2023.

[16] Q. Aumann, P. Benner, I. V. Gosea, J. Saak, and J. Vettermann, "A tangential interpolation framework for the AAA algorithm," *Proc. Appl. Math. Mech.*, vol. 23, no. 3, p. e202300183, 2023.

[17] A. Hochman, "FastAAA: A fast rational-function fitter," in *2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2017, pp. 1–3.

[18] P. Lietaert, K. Meerbergen, J. Pérez, and B. Vandereycken, "Automatic rational approximation and linearization of nonlinear eigenvalue problems," *IMA J. Numer. Anal.*, vol. 42, no. 2, pp. 1087–1115, 2022.

[19] S. Elsworth and S. Güttel, "Conversions between barycentric, RKFUN, and Newton representations of rational interpolants," *Lin. Alg. Appl.*, vol. 576, pp. 246–257, 2019.

[20] S. Güttel, D. Kressner, and B. Vandereycken, "Randomized sketching of nonlinear eigenvalue problems," *arXiv preprint arXiv:2211.12175*, 2022.

[21] T. Driscoll, Y. Nakatsukasa, and L. N. Trefethen, "AAA rational approximation on a continuum," *arXiv preprint arXiv:2305.03677*, 2023.

[22] A. Ilchmann, *Non-identifier-based high-gain adaptive control*. Springer, 1993, vol. 189.

[23] B. Brogliato, B. Maschke, R. Lozano, and O. Egeland, *Dissipative systems analysis and control*, ser. Communications and Control Engineering (CCE). Springer, 2007, vol. 2.

[24] T. Bradde, S. Grivet-Talocia, Q. Aumann, and I. V. Gosea, "A modified AAA algorithm for learning stable reduced-order models from data," 2023, arXiv, 2312.16978, math.NA.

[25] J. Taylor, "Strictly positive-real functions and the Lefschetz-Kalman Yakubovich (LKY) lemma," *IEEE Trans. Circuits Syst.*, vol. 21, no. 2, pp. 310–311, 1974.

[26] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *In Proc. of the CACSD Conference*, Taipei, Taiwan, 2004.

[27] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 10.1.16.*, 2023. [Online]. Available: https://docs.mosek.com/latest/toolbox/index.html

[28] "IdEM: Electronic device characterization," 2023. [Online]. Available: https://www.3ds.com/products/simulia/idem