High-Performance Passive Macromodeling Algorithms for Parallel Computing Platforms

High-Performance Passive Macromodeling Algorithms for Parallel Computing Platforms

Alessandro Chinea, Stefano Grivet-Talocia, *Senior Member, IEEE*, Salvatore B. Olivadese and Luca Gobbato,

***Abstract*—This work presents a comprehensive strategy for fast generation of passive macromodels of linear devices and interconnects on parallel computing hardware. Starting from a raw characterization of the structure in terms of frequency-domain tabulated scattering responses, we perform a rational curve fitting and a postprocessing passivity enforcement. Both algorithms are parallelized and cast in a form that is suitable for deployment on shared-memory multicore platforms. Particular emphasis is placed on the passivity characterization step, which is performed using two complementary strategies. The first uses an iterative restarted and deflated rational Arnoldi process to extract the imaginary Hamiltonian eigenvalues associated to the model. The second is based on an accuracy-controlled adaptive sampling. Various parallelization strategies are discussed for both schemes, with particular care on load balancing between different computing threads and memory occupation. The resulting parallel macromodeling flow is demonstrated on a number of medium and large scale structures, showing good scalability up to 16 computational cores.**

***Index Terms*—Adaptive sampling, Linear macromodeling, Passivity, Hamiltonian matrices, Perturbation theory, Eigenvalues, Singular Values, Scattering, Parallel algorithms.**

I. Introduction

The verification of Signal and Power Integrity of digital, mixed-signal and RF systems is almost invariably based on time-domain system-level numerical simulations. The latter are often run using circuit solvers of the SPICE class, applied to netlists that include suitable circuit models of all system parts, including both linear devices such as filters, baluns, connectors, vias and interconnects in general, and nonlinear devices such as drivers, receivers, amplifier circuits, etc. The presence of the nonlinear devices is the main reason why the system simulation has to be conducted in the time domain [1], [2].

Since the electrical performance of interconnect and linear devices in general is best represented in the frequency domain, it is a standard practice to characterize such structures through a representative set of frequency samples of their transfer matrix in scattering, impedance or admittance form, suitably spread over the frequency band of interest, possibly obtained from direct measurement or full-wave simulations. The conversion of this representation into circuit models ready to be plugged in a system-level SPICE netlist is thus necessary. The standard approach to achieve this goal is to resort to the many available macromodeling techniques that have appeared over the last few years [3]-[40]. In this work, we intend the term macromodeling as a set of methods that extract a simulation model, i.e., a set of equations typically but not necessarily in state-space form, starting from a finite set of input-output data samples.

The most successful macromodeling approaches are based on two key steps. First, an initial model is identified through a curve-fitting stage [3]-[14]. A closed-form representation of the model as a rational transfer matrix in the Laplace domain is obtained by determining the model parameters (i.e., poles and residues) that minimize some least squares error with respect to the available frequency samples. This representation is easily cast as a state-space realization [41]. A second step checks the model and, if needed, corrects the model coefficients by enforcing its passivity [15]-[40]. This second step is essential to avoid possible instabilities in the transient system-level simulations, as discussed in [42]-[44]. Finally, the synthesis of the resulting state-space realization as a SPICE-ready equivalent circuit is straightforward [2].

Despite the above macromodeling flow is now well established, there may be some difficulties in its application to medium and large scale complex structures. Here, complexity is intended as a collective measure of the amount of computations that are required for the model generation. This measure is influenced by the number of available frequency samples, by the number of ports of the device of interest, and by the dynamic order, i.e., the size of a minimal state-space realization [41], that is required for an accurate representation of the system dynamics over the bandwidth of interest. In this work, we present a set of techniques that may be used to drastically reduce the total model extraction time by means of parallelization and deployment on multicore hardware architectures. It is widely acknowledged that high-performance numerical schemes will have to take advantage of massively parallel computers, that are expected to become ubiquitous in the next few years even at the desktop level.

A parallelization strategy for rational curve fitting has been presented as the Parallel Vector Fitting (PVF) scheme in [14]. So we only give a brief outline of the algorithm here, in order to support and justify the numerical results. Some preliminary work on parallelization of passivity enforcement schemes has also been presented in [40], where a parallel eigensolver for Hamiltonian matrices with sparse structure or admitting a sparse factorization was presented. In this work,

Manuscript received ; revised.

This work was supported in part by the Italian Ministry of University (MIUR) under a Program for the Development of Research of National Interest (PRIN grant #2008W5P2K).

Alessandro Chinea is with IdemWorks s.r.l., Torino, Italy (e-mail: a.chinea@idemworks.com); Stefano Grivet-Talocia and Salvatore B. Olivadese are with the Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy (e-mail: salvatore.olivadese@polito.it, stefano.grivet@polito.it); Luca Gobbato is with the Department of Control and Computer Engineering, Politecnico di Torino, Torino, Italy (e-mail: luca.gobbato@polito.it);

we further extend this approach and we complement it with a parallel adaptive sampling scheme, presented as an extension of the singular value/vector tracking scheme of [28]. The main contribution of this work is the reformulation and the parallelization of these schemes in order to achieve a good scalability when increasing the number of computational cores. This is in general a difficult task due to the presence of unavoidable serial content or synchronization points in the code that may limit its performance significantly. We show that the overall macromodeling flow that is obtained is capable of maintaining a very good speedup factor up to 16 computational cores, the maximum currently available on our largest server. In terms of runtime, complex interconnect models required by Signal and Power Integrity analyses are extracted in seconds or in the worst case in few minutes. These results make the computational cost of the model extraction step essentially negligible with respect to the cost of other verification steps, including the preceding electromagnetic simulations or the subsequent circuit analysis.

This paper is organized as follows. Section II introduces the problem, sets notation, and provides background information. Section III reviews the adaptive sampling scheme of [28] and presents the proposed parallelization strategy. Section IV reviews and discusses the parallelization of Hamiltonian-based passivity checks. Section V discusses possible strategies for combining the Hamiltonian-based passivity check with the adaptive sampling process in order to maximize performance. Passivity enforcement and its parallelization is addressed in Sec. VI. Finally, Section VII reports and discusses the numerical results obtained on a significant set of benchmarks.

Throughout this paper, *, $^\mathsf{T}$ and $^\mathsf{H}$ stand for complex conjugate, transpose, and conjugate (hermitian) transpose, respectively. The sets of eigenvalues and singular values of a complex matrix $\boldsymbol{X}$ are denoted as $\lambda(\boldsymbol{X})$ and $\sigma(\boldsymbol{X})$, respectively. The maximum singular value is denoted as $\sigma_{\max}(\boldsymbol{X}) = \|\boldsymbol{X}\|$. The Frobenius norm of matrix $\boldsymbol{X}$ is defined as $\|\boldsymbol{X}\|_F^2 = \sum_{ij} |X_{ij}|^2$. The operators $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ round their argument to the nearest larger and smaller integer, respectively.

II. Background

A. Frequency-domain macromodel extraction

We consider the problem of frequency-domain macromodel extraction. The starting point is a set of frequency samples

$$(\omega_k, \hat{\boldsymbol{H}}_k), \quad k = 1, \ldots, K\,, \tag{1}$$

where the complex $P \times P$ matrix $\hat{\boldsymbol{H}}_k = \hat{\boldsymbol{H}}(\jmath\omega_k)$ is the original computed or measured response of the structure under investigation at frequency $\omega = \omega_k$. A macromodel is here defined as the state-space form

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{x}}(t) &=& \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)\,, \\ \boldsymbol{y}(t) &=& \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t)\,, \end{array} \right. \tag{2}$$

with $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $\boldsymbol{B} \in \mathbb{R}^{N\times P}$, $\boldsymbol{C} \in \mathbb{R}^{P\times N}$, $\boldsymbol{D} \in \mathbb{R}^{P\times P}$, with corresponding transfer matrix

$$\boldsymbol{H}(s) = \boldsymbol{D} + \boldsymbol{C}\left(s\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{B}\,, \tag{3}$$

where s is the Laplace variable. We will focus our attention to systems in scattering representation, so that (2) and (3) are scattering matrices associated to some prescribed port resistances $R_{0i} > 0$ for $i = 1, \ldots, P$. However, all material in this paper applies to other input-output representations, including impedance, admittance or more general hybrid forms, with obvious modifications.

The identification of the state-space matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ from the data samples is typically performed using a rational curve fitting process that finds poles p_j and residue matrices $\boldsymbol{R}_j$ of the following partial fraction form

$$\boldsymbol{H}(s) = \boldsymbol{R}_\infty + \sum_{j=1}^{n} \frac{\boldsymbol{R}_j}{s - p_j} \tag{4}$$

by solving the following optimization problem

$$\min_{p_j, \boldsymbol{R}_j, \boldsymbol{R}_\infty} \sum_{k=1}^{K} \left\| \boldsymbol{R}_\infty + \sum_{j=1}^{n} \frac{\boldsymbol{R}_j}{\jmath\omega_k - p_j} - \hat{\boldsymbol{H}}_k \right\|_F^2, \tag{5}$$

where the approximation error is measured in the Frobenius norm. The conversion of (4) into a state-space realization (2) is straightforward and will not be discussed here [41]. We only remark that, in case the rank of the residue matrices $\boldsymbol{R}_j$ is full for each j, the size of the state-space matrix $\boldsymbol{A}$ of a minimal realization is $N = nP$. In the following, we will assume a Gilbert realization [24], [45], with $\boldsymbol{A}$ block-diagonal with blocks of size 1 for the synthesis of real poles and of size 2 for complex pole pairs. Note that this choice is always possible in the construction of the state-space realization and leads to a $O(N)$ computational cost for the evaluation of the transfer matrix at a given frequency.

The Vector Fitting (VF) scheme [8]-[14] is the most prominent rational curve fitting tool for the solution of (5). As discussed in [8], the VF scheme reformulates the non-convex form (5) as an iterative sequence of simple linear least squares (LS) problems, associated to a pole relocation stage that iteratively refines a set of initial poles. Although no theoretical proof of convergence is available, except for trivial cases, experience shows that the performance of the VF scheme is excellent, making this method the tool of choice for macromodel extraction.

It has been shown in [14] that the computational cost of the VF scheme in its most advanced formulation [11] scales approximately as $O(P^2n^2K)$ per iteration. This cost is dominated by a QR factorization step required by the LS solution [46]. This implies that macromodel extraction for structures characterized by a large number of ports and requiring a large number of poles may require significant computational resources and overall runtime. Two main attempts towards the reduction of this cost have been reported. In [13], a preprocessing data reduction step is applied to the samples (1), resulting in a small number $\rho \ll P^2$ of frequency-dependent "basis" functions $\varphi_\nu(\jmath\omega_k)$. With a full control over the approximation error, each response $\hat{\boldsymbol{H}}_{i,j}(\jmath\omega_k)$ in the original dataset is expressed as a linear combination of these basis functions. Therefore, a macromodel is obtained by applying VF to the reduced basis, with a cost reduction by a factor ρ/P^2, plus an overhead due to the initial data compression stage. Full details on this procedure are available in [13].

A second approach to computational cost reduction of VF is through parallelization. The Parallel Vector Fitting (PVF) scheme, presented in [14], shows that excellent performance is possible when restructuring the various steps of the basic VF algorithm so that most operations can be performed concurrently by a number of T computational threads running on a shared memory computer. The parallel efficiency of the PVF scheme in its various implementations is nearly ideal for large-scale cases. A detailed presentation of PVF and its performance is available in [14], so we will not discuss this scheme further, although we will use it in Section VII to present scalability results of the complete macromodel extraction flow.

B. Passivity constraints

In order to be useful for subsequent system-level (transient) simulations, the macromodel (2) must be compliant with fundamental physics-based constraints: causality, stability, and passivity. For rational macromodels, causality and stability are guaranteed by the unique condition that all model poles should have a negative real part, $\Re p_j < 0$, $\forall j$, see e.g. [42]. This condition is easily enforced during the VF pole relocation stage. Model passivity is more difficult to guarantee, and a special set of constraints and associated enforcement algorithms are necessary.

The fundamental condition under which a (scattering) transfer matrix $\boldsymbol{H}(s)$ represents a passive macromodel is bounded realness[1] [42], [44]. A transfer matrix $\boldsymbol{H}(s)$ is *Bounded Real (BR)* if

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

The first two conditions are guaranteed if the state-space realization (2) is real-valued and asymptotically stable [41], [47]. Under these assumptions, the last condition can be relaxed and checked only on the imaginary axis $s = \jmath\omega$,

$$\boldsymbol{\Theta}(\jmath\omega) \geq 0\,, \quad \forall\omega\,, \tag{6}$$

which in turn is equivalent to requiring that all singular values of $\boldsymbol{H}(\jmath\omega)$ must be uniformly bounded by one at any frequency

$$\sigma_i \leq 1\,, \quad \forall\sigma_i \in \sigma(\boldsymbol{H}(\jmath\omega))\,, \quad \forall\omega\,. \tag{7}$$

Note that $\sigma_i = \sqrt{1-\lambda_i}$, where $\lambda_i \in \lambda(\boldsymbol{\Theta}(\jmath\omega))$ are the eigenvalues of $\boldsymbol{\Theta}(\jmath\omega)$, so that (7) is equivalent to

$$\lambda_i \geq 0\,, \quad \forall\lambda_i \in \lambda(\boldsymbol{\Theta}(\jmath\omega))\,, \quad \forall\omega\,. \tag{8}$$

The passivity condition (7) should be checked for each frequency $\omega \in \mathbb{R}$. This suggests using a frequency sampling process to extract a significant set of frequency points ω_k and to perform (7) on these samples only. In order for this approach to be reliable, the set of samples ω_k must be determined adaptively based on the dynamic features of the macromodel, in order to avoid missing important information due to an incomplete characterization. One of the main contributions of this work is indeed a parallel scheme for an adaptive and accuracy-controlled sample extraction, finalized at the fast execution of passivity check (7). This scheme is presented in Sec. III.

The passivity condition (7) includes also the asymptotic value $\omega \to \infty$

$$\boldsymbol{\Theta}(\infty) = \boldsymbol{I} - \boldsymbol{D}^{\mathsf{T}}\boldsymbol{D} \geq 0 \quad \Leftrightarrow \quad \|\boldsymbol{D}\| = \sigma_{\max}(\boldsymbol{D}) \leq 1\,. \tag{9}$$

If the slightly stronger condition

$$\|\boldsymbol{D}\| = \sigma_{\max}(\boldsymbol{D}) < \alpha < 1 \tag{10}$$

with a suitable constant α is enforced during the construction of the macromodel, then a passivity check can be performed through the *Hamiltonian matrix* associated to the macromodel [48], [49], defined for the scattering form as

$$\mathcal{M} = \begin{pmatrix} \boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{D}^{\mathsf{T}}\boldsymbol{C} & -\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathsf{T}} \\ \boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\boldsymbol{C} & -\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{C}^{\mathsf{T}}\boldsymbol{D}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathsf{T}} \end{pmatrix}, \tag{11}$$

where $\boldsymbol{R} = (\boldsymbol{D}^{\mathsf{T}}\boldsymbol{D} - \boldsymbol{I})$ and $\boldsymbol{S} = (\boldsymbol{D}\boldsymbol{D}^{\mathsf{T}} - \boldsymbol{I})$. It is well known [23], [48] that the frequencies $\bar{\omega}_k$ at which one of the singular values σ_i reaches the threshold $\gamma = 1$ correspond to the purely imaginary eigenvalues $\mu_k = \jmath\bar{\omega}_k$ of the Hamiltonian matrix $\mathcal{M}$. This fact can be exploited [23] to devise a simple scheme for the determination of the frequency bands where the passivity condition (7) is violated.

This procedure does not require any sampling but involves the determination of the eigenspectrum of the Hamiltonian matrix. This operation requires $O(N^3)$ operations, which might be impractical for large-scale models. However, when the state-space realization of the macromodel is sparse (as in our case), it is possible to apply a Krylov subspace projection [50]-[54] to the Hamiltonian eigenproblem, in order to extract the few eigenvalues of interest in a prescribed region of the complex plane. This fact has been exploited in [24], where a multishift restarted Arnoldi process [50]-[52], [55] similar to the Complex Frequency Hopping (CFH) scheme [56] has been applied to extract all purely imaginary Hamiltonian eigenvalues. Also this scheme is an excellent candidate for parallelization, as discussed in [40]. This paper extends the preliminary results of [40] by presenting in Sec. IV a dedicated dynamic scheduling that reduces thread-dependency and improves parallel efficiency. Parallel adaptive sampling and parallel Hamiltonian eigensolution will be combined into a single algorithm in Sec. V

Before presenting the details of each scheme, we depict in Fig. 1 the results that are expected at the end of the passivity check. There are two main quantities of interest:

- the intersections $\bar{\omega}_k$ of the singular value trajectories with the threshold $\gamma = 1$ (square dots in the figure); these are directly obtained from the Hamiltonian eigenvalues;
- the local maxima $(\hat{\omega}_k, \hat{\sigma}_k)$ of the singular values within each frequency band with passivity violations (black dots in the figure). These maxima can be obtained both by adaptive sampling and by repeated computation of imaginary eigenvalues of suitably modified Hamiltonian matrices [23], [48].

[1] In case $\boldsymbol{H}(s)$ is an impedance or admittance matrix, the bounded realness conditions should be replaced by *positive realness* conditions, simply obtained by redefining $\boldsymbol{\Theta}(s) = \boldsymbol{H}(s) + \boldsymbol{H}(s)^{\mathsf{H}}$.

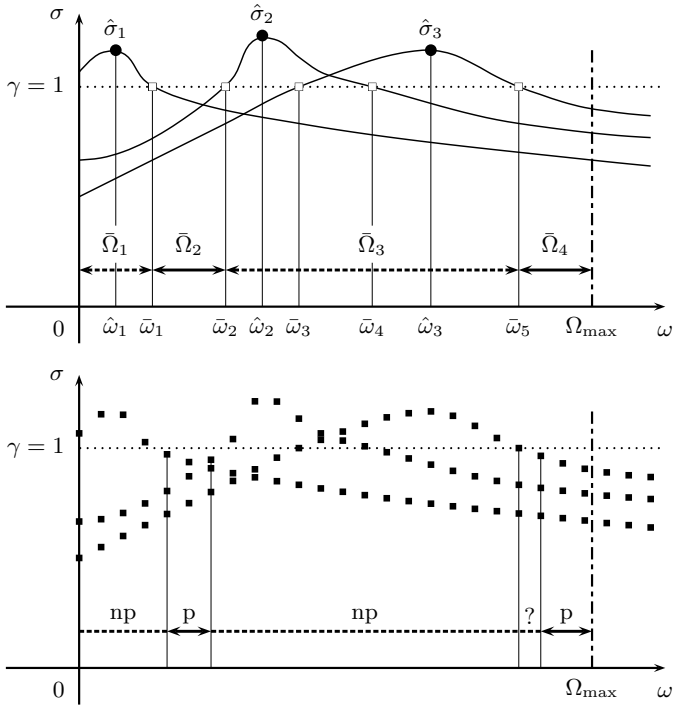

Fig. 1. Graphical illustration of alternative passivity characterizations. Top: Hamiltonian-based check, with precise determination of passive/nonpassive frequency bands. Bottom: passivity check based on frequency sampling. See text for details.

One of the main objectives of this work is to determine the optimal strategy for the computation of the above quantities in the least time using T concurrent computational threads on a multicore computer.

III. Parallel adaptive sampling

The main objective of the proposed Parallel Adaptive Sampling (PAS) scheme is to determine a partition of the frequency axis $\Omega = [0, \infty)$ into disjoint subbands

$$\Omega = \bigcup_{q=1}^{Q} \Omega_q \,, \qquad \Omega_q = [\omega_{q-1}, \omega_q) \tag{12}$$

with $\omega_0 = 0$ and $\omega_Q = +\infty$. Defining the interior of each subband as

$$\check{\Omega}_q = (\omega_{q-1}, \omega_q) = \Omega_q - \{\omega_{q-1}\} \,, \tag{13}$$

the partition (12) is determined such that one of the following conditions will hold for each subband $\check{\Omega}_q$

- $\max_i \sigma_i(\jmath\omega) > 1$, $\forall \omega \in \check{\Omega}_q$: in this case, passivity condition (7) is violated at any point within the subband, which is thus flagged as "non-passive" with the superscript $^{\rm np}$.
- $\max_i \sigma_i(\jmath\omega) < 1$, $\forall \omega \in \check{\Omega}_q$: in this case, (7) holds at any point within the subband, which is thus flagged as "passive" with the superscript $^{\rm p}$.
- $\max_i \sigma_i(\jmath\omega) \approx 1$, $\forall \omega \in \check{\Omega}_q$: in this case, the maximum singular value will be too close to the threshold $\gamma = 1$ in order to qualify the system as locally passive or non-passive in $\check{\Omega}_q$. We want to make sure that this last case is such that $|\Omega_q| = \omega_q - \omega_{q-1}$ is small. This undetermined case will be flagged with the superscript $^{?}$.

Passive, non-passive, and undetermined bands will be collected as

$$\begin{aligned} \Omega^{\rm np} &= \textstyle\bigcup_q \Omega_q : \quad \max_i \sigma_i(\jmath\omega) > 1, \, \forall \omega \in \check{\Omega}_q \\ \Omega^{\rm p} &= \textstyle\bigcup_q \Omega_q : \quad \max_i \sigma_i(\jmath\omega) < 1, \, \forall \omega \in \check{\Omega}_q \\ \Omega^{?} &= \textstyle\bigcup_q \Omega_q : \quad \Omega_q \not\subseteq \Omega^{\rm np} \cup \Omega^{\rm p} \end{aligned} \tag{14}$$

In addition, for each non-passive subband $\Omega_q \subseteq \Omega^{\rm np}$, we want to find all local maxima $\hat{\sigma}_k$ and the corresponding frequencies $\hat{\omega}_k$ at which these maxima are attained. See Fig. 1 for a graphical illustration.

A. Accuracy-controlled sampling via eigenvector tracking

We recall that, since $\boldsymbol{A}$ has no purely imaginary poles, the singular values $\sigma_i(\jmath\omega)$ are continuous and differentiable functions of frequency [57]. However, when computing these singular values numerically over a prescribed discrete set of frequencies $\{\omega_k\}$, there is no guarantee that each $\sigma_i(\jmath\omega_k)$ for fixed i collects samples from the *same* singular value trajectory. The computation at each frequency ω_k is in fact independent, and the adopted singular value or eigenvalue solver may return its results with an order that may differ from one sample to the next. Especially when two singular value trajectories cross at some frequency, the tracking becomes ambiguous.

The first objective is thus to dynamically determine a set of frequencies $\{\omega_k\}$ that is sufficient to track the individual smooth singular value trajectories by a suitable reordering. This reordering can be achieved by a mode tracking scheme [58], such as the one presented in [28]. Given two available (adjacent) frequency samples ω_m and ω_{m+1}, we compute the eigendecomposition of $\boldsymbol{\Theta}(\jmath\omega_m)$ and $\boldsymbol{\Theta}(\jmath\omega_{m+1})$ and we collect the eigenvalues into matrices $\boldsymbol{\Lambda}_m$ and $\boldsymbol{\Lambda}_{m+1}$ and the (orthogonal and unit-normalized) eigenvectors into matrices $\boldsymbol{V}_m$ and $\boldsymbol{V}_{m+1}$. Note that these matrices coincide with the right singular vectors of $\boldsymbol{H}(\jmath\omega)$. Then, we compute all possible mutual scalar products among all these eigenvectors as entries in matrix

$$\widetilde{\boldsymbol{P}}_{m,m+1} = \boldsymbol{V}_m^{\mathsf{H}} \boldsymbol{V}_{m+1} \,. \tag{15}$$

If the two frequencies are sufficiently close so that the direction of the eigenvectors undergoes a small change from ω_m to ω_{m+1}, then $\widetilde{\boldsymbol{P}}_{m,m+1}$ will have approximately the structure of a permutation matrix, with one single element per row and column with magnitude close to 1, and with all other elements nearly 0. If this is true, the permutation matrix $\boldsymbol{P}_{m,m+1}$ that reorders the eigenvectors and eigenvalues from sample m to sample $m+1$ is obtained by rounding the magnitude of each element of $\widetilde{\boldsymbol{P}}_{m,m+1}$ towards 0 or 1. A numerical test whether this tracking/permutation is successful can be obtained by checking

$$\max_{i,i'} \left\{ \left| \left(|\boldsymbol{P}^{\mathsf{T}}_{m,m+1} \widetilde{\boldsymbol{P}}_{m,m+1}| - \boldsymbol{I} \right) \right|_{i,i'} \right\} < \varepsilon \tag{16}$$

for a suitable threshold $\varepsilon \ll 1$. We refer to [28] for more details. If condition (16) is fulfilled, we infer that the behavior

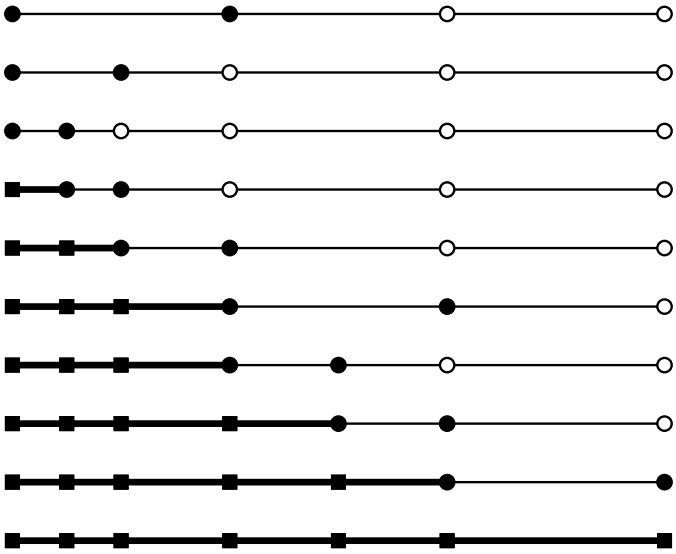

Fig. 2. Adaptive frequency sampling via local refinement (serial implementation). Each row from top to bottom corresponds to one application of the $\mathcal{R}$ check (17). White dots denote samples still to be processed. Black dots denote samples being used by current $\mathcal{R}$ check. Black squares denote samples that do not need any more processing. A thick line highlights a frequency band that is finalized and which does not need further refinement.

of the system transfer function and its singular values is well resolved within $[\omega_m, \omega_{m+1}]$. Otherwise, a new sample $\omega_{m+1/2} = (\omega_m + \omega_{m+1})/2$ is added and the check is applied again to the two subintervals $[\omega_m, \omega_{m+1/2}]$ and $[\omega_{m+1/2}, \omega_{m+1}]$. Binary subdivision of each pair of adjacent samples drawn from an initial distribution is applied recursively until (16) is met everywhere.

B. Parallel Adaptive Sampling

Let us take a closer look at the above described adaptive refinement scheme. Formally, the refinement check is expressed as

$$\nu = \mathcal{R}(\omega_m, \omega_{m+1}) \,, \tag{17}$$

where the input arguments define the local band to be checked, and the output ν can be either $\omega_{m+1/2}$ or the empty set $\emptyset$, in which case no further refinement is required. Evaluation of (17) requires the computation of transfer matrix $\boldsymbol{H}(\jmath\omega)$ at the two frequencies ω_m, ω_{m+1}, together with its right singular vector matrices $\boldsymbol{V}_m$ and $\boldsymbol{V}_{m+1}$. As part of the $\mathcal{R}$ check, we include the following computations: if ν is empty, the resulting permutation matrix $\boldsymbol{P}_{m,m+1}$ is immediately applied to reorder the singular values at ω_{m+1}; otherwise, the new sample $\omega_{m+1/2}$ is computed together with its associated transfer matrix $\boldsymbol{H}(\jmath\omega_{m+1/2})$ and singular vector matrix $\boldsymbol{V}_{m+1/2}$, which are stored for the next check.

Iterative application of (17) determines a binary subdivision tree of the frequency axis, where each node in the tree denotes a frequency sample. Figure 2 illustrates the order in which the $\mathcal{R}$ check is applied in a serial implementation, where we assumed that the leftmost local subband that is still to be refined is processed first. The figure shows that the subbands are finalized starting from the left edge of the initial frequency interval. This consideration leads to a simple strategy for the

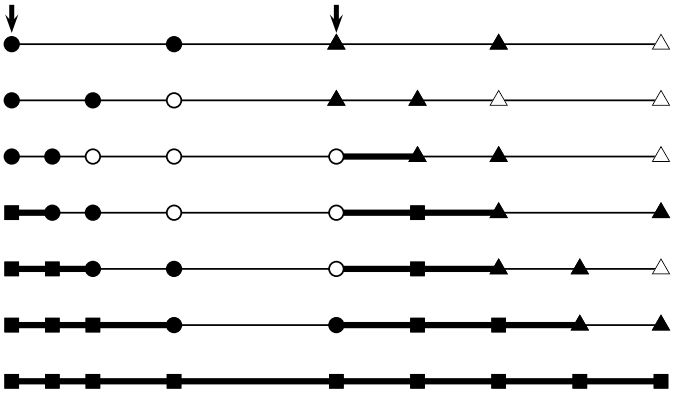

Fig. 3. Parallel adaptive frequency sampling via local refinement using $T = 2$ threads. Samples assigned to thread $t = 1$ ($t = 2$) are depicted with circles (triangles). Arrows indicate start points (leftmost sample) for the two threads. White fill denotes samples still to be processed, whereas black fill denotes samples used by current iteration. Black squares denote samples that do not need any more processing. A thick line highlights a frequency band that is finalized and which does not need further refinement.

parallelization of this refinement scheme using T concurrent threads, based on the following steps and rules.

1) Startup: At startup, a set of initial frequency samples $\mathcal{S}^0$ is determined. Here, we form this set as the union of samples obtained independently through different strategies:

- an upper frequency $\Omega_{\max}$ is determined following the procedure in [24], with the guarantee that no passivity violations occur for $\omega > \Omega_{\max}$; therefore, only the interval $[0, \Omega_{\max}]$ needs to be checked instead of the full imaginary axis;
- a set $\mathcal{S}_{\lin}$ of $k_{\lin}$ uniformly spaced samples are determined in $[0, \Omega_{\max}]$, including edges;
- a set $\mathcal{S}_{\log}$ of logarithmically spaced samples with k_d samples per decade are computed from $\omega_{\min}$ to $\omega_{\max}$, where k_d, $\omega_{\min}$ and $\omega_{\max}$ depend on the particular application and structure of interest;
- a set $\mathcal{S}_p$ of samples is obtained as in [28] from the model poles $p_i = \alpha_i \pm \jmath\beta_i$ by sampling uniformly with $2R+1$ points the phase of the associated resonance curve, as

$$\mathcal{S}_p = \bigcup_{i,r} \left\{ \omega_{i,r} = \beta_i + \alpha_i \tan \frac{r\pi}{2(R+1)} \right\} \tag{18}$$

  with $r = -R, \ldots, R$.

As a result, the set of initial samples that will be subject to the $\mathcal{R}$ iteration is defined as

$$\mathcal{S}^0 = \mathcal{S}_{\lin} \cup \mathcal{S}_{\log} \cup \mathcal{S}_p \,, \tag{19}$$

with all samples reordered for increasing values.

2) Initial workload allocation: Supposing that we have T threads that can operate concurrently, we partition the set of initial samples as

$$\mathcal{S}^0 = \bigcup_{t=1}^{T} \mathcal{S}_t^0 \,, \tag{20}$$

where the number of elements of each subset is $\#\{\mathcal{S}_t^0\} = \lfloor \#\{\mathcal{S}^0\}/T \rfloor$ for $t = 0, \ldots, T-1$. The remaining samples

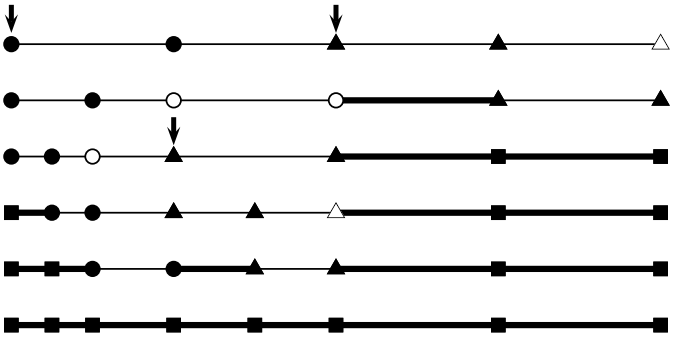

Fig. 4. Parallel adaptive frequency sampling via local refinement using $T = 2$ threads and dynamic rescheduling (same notation as in Fig. 3). Note that thread $t = 2$ is restarted at the third iteration after completing its initially assigned workload.

are assigned to $\mathcal{S}_T^0$. The subdivision is ordered, such that for $t_1 < t_2$,

$$\forall \omega_i \in \mathcal{S}_{t_1}^0 \quad \text{and} \quad \forall \omega_j \in \mathcal{S}_{t_2}^0 \quad \Rightarrow \quad \omega_i \leq \omega_j \,, \tag{21}$$

with each pair of adjacent subbands $\mathcal{S}_{t_i}^0$ and $\mathcal{S}_{t_{i+1}}^0$ sharing the single sample

$$\tilde{\omega}_i = \max \mathcal{S}_{t_i}^0 = \min \mathcal{S}_{t_{i+1}}^0 \,. \tag{22}$$

Each subset $\mathcal{S}_t^0$ is allocated statically to thread t, which iteratively applies the $\mathcal{R}$ refinement check until the entire subband is covered, as in Fig. 2. This initial allocation ensures that, if no refinement is required, approximately the same amount of work is allocated for each thread. Figure 3 illustrates this process, showing the evolution of each subset of samples $\mathcal{S}_t^\nu$ at few iterations ν. In the following, we will drop the iteration count ν.

3) Dynamic thread reallocation: As the iterative refinement check proceeds and each subband is processed independently by each thread, it may happen that some bands require more adaptive refinement steps than others. Therefore, it may happen that one thread t_j completes its refinement task when the other threads are still working. In this case, we should avoid leaving the thread inactive, since this would compromise parallel efficiency. In order to find some work to do for the idle thread t_j, we scan the remaining threads t_i for $i \neq j$ and we determine the number of sample pairs in set $\mathcal{S}_{t_i}$ that at current iteration are still to be processed by the $\mathcal{R}$ check. Although it is not guaranteed that the work for these threads will coincide with the corresponding number of unchecked subbands, the number of expected $\mathcal{R}$ iterations will not certainly be smaller. Therefore, we identify the thread t_ℓ that requires the largest amount of estimated $\mathcal{R}$ checks and we restart thread t_j by assigning to it one half of the samples still to be processed by t_ℓ. More precisely, we split

$$\mathcal{S}_\ell \rightarrow \hat{\mathcal{S}}_{t_\ell} \cup \hat{\mathcal{S}}_{t_j} \tag{23}$$

with the constraint

$$\forall \omega_i \in \hat{\mathcal{S}}_{t_\ell} \quad \text{and} \quad \forall \omega_k \in \hat{\mathcal{S}}_{t_j} \quad \Rightarrow \quad \omega_i \leq \omega_k \,, \tag{24}$$

with the two sets $\hat{\mathcal{S}}_{t_\ell}$, $\hat{\mathcal{S}}_{t_j}$ sharing only one sample. This strategy guarantees an initially equal subdivision of the workload between t_j and t_ℓ. Figure 4 provides a graphical illustration of this thread reallocation. Then, the thread reallocation process is repeated anytime some thread becomes idle, by rescheduling it to help the most busy thread at that time.

4) End of refinement pass: The above described multi-thread adaptive refinement process stops when all threads have completed their tasks. Due to the proposed optimized dynamic scheduling, the algorithm is automatically load balanced, except for the last iteration during which a group of threads might remain idle while the other threads are completing their last task. The maximum total duration of this last step is the time required for a single $\mathcal{R}$ iteration.

In addition to the natural stopping condition for the $\mathcal{R}$ iteration, which occurs when $\emptyset$ is returned by (17) and in which case all singular value trajectories are tracked based on their singular vector perturbation, we add an additional stopping condition in terms of the maximum number of nested refinements $I_{\max}$. This parameter intervenes when tracking is not possible, e.g., in the case of singular values with higher multiplicity, whose singular vectors cannot be defined uniquely. In all numerical tests in this paper we used $I_{\max} = 6$, which was observed to provide a good compromise between accuracy and efficiency.

C. Local passivity check

The final result of the above refinement scheme is a set of frequency samples ω_k and a reordered sequence (through the above-defined permutation matrices $\boldsymbol{P}_{m,m+1}$) of singular values samples. For fixed i, the reordered samples $\sigma_i(\jmath\omega_k)$ can thus be considered to be drawn from a continuous and differentiable trajectory $\sigma_i(\jmath\omega)$. Exploitation of this smoothness leads to various straightforward ways of checking passivity between each pair of adjacent frequencies. One can define a worst-case linear prediction error at sample ω_m based on a first-order eigenvalue perturbation from the adjacent left and right samples [28]

$$\Delta_m^\pm = \max_i \left\{ \left| \left(\boldsymbol{V}_{m\pm1}^{\mathsf{H}} \boldsymbol{\Theta}_m \boldsymbol{V}_{m\pm1} \right)_{ii} - (\boldsymbol{\Lambda}_m)_{ii} \right| \right\} , \tag{25}$$

and infer that the model is locally passive in a neighborhood of ω_m if

$$\max_i \sigma_i(\jmath\omega_m) + \beta \max\{\Delta_m^-, \Delta_m^+\} < 1 \,, \tag{26}$$

where $\beta > 1$ is a parameter used to compensate for the missing higher order terms in the linear prediction (see [28] for details). This local check at ω_m can be formally expressed as

$$\vartheta_m = \mathcal{C}(\omega_{m-1}, \omega_m, \omega_{m+1}) \,, \tag{27}$$

where ϑ_m is either 0 (flagging locally non-passive samples) or 1 (locally passive samples), since a symmetric check is performed using both samples at the left and right of current sample. The only exception is when the check is performed at the edge of the bandwidth of interest, in which case only two samples are used to construct a one-sided linear prediction error Δ_m^- or Δ_m^+.

Performing this local passivity check using T computational threads is straightforward, since a direct static scheduling

is sufficient. In fact, since the $\mathcal{C}$ check is performed on a prescribed set of samples which remains fixed and does not grow through iterations, the static work allocation discussed in Sec III-B2 is already optimal. Therefore, we do not discuss this aspect further.

As a result from above procedure, the model is concluded to be passive in (ω_m, ω_{m+1}) if (26) is satisfied at both ω_m and ω_{m+1}. Conversely, the model is concluded to be non-passive in (ω_m, ω_{m+1}), or at least in some portion of it, if any of the maximum singular values at sample m and $m+1$ is larger than one,

$$\max_i \sigma_i(\jmath\omega_m) > 1 \quad \text{or} \quad \max_i \sigma_i(\jmath\omega_{m+1}) > 1\,. \tag{28}$$

For all other cases in which

$$\max_i \sigma_i(\jmath\omega_m) \leq 1 \quad \text{and} \quad \max_i \sigma_i(\jmath\omega_{m+1}) \leq 1\,, \tag{29}$$

but (26) is not satisfied at ω_m and ω_{m+1}, the subband is flagged as undetermined since the singular value trajectories are too close to the threshold.

Once all subbands are flagged, adjacent passive (nonpassive or undetermined) bands are merged to form the subdivision (12). Finally, the local maxima $(\hat{\omega}_k, \hat{\sigma}_k)$ of the singular value trajectories for each nonpassive subband are determined by constructing a local quadratic polynomial that interpolates three adjacent samples and by taking its peak value. All these operations require negligible time and are performed as a serial postprocessing in our implementation.

D. Optimizations

The local passivity check $\mathcal{C}$ as described above is performed after the adaptive refinement iteration $\mathcal{R}$ is completed. This strategy presents some critical aspects related to memory use and management. In fact, the $\mathcal{C}$ check requires to store, for each sample ω_m to be checked, the matrix $\boldsymbol{\Theta}(\jmath\omega_m)$, the eigenvalue matrix $\boldsymbol{\Lambda}_m$, and the eigenvector matrices at the left and right samples $\boldsymbol{V}_{m\pm1}$. So, until a subband (ω_m, ω_{m+1}) is definitely flagged as passive/non-passive/undetermined, all the above quantities need to be stored for each of the two samples m, $m+1$. For a $P \times P$ transfer function resulting into a number K of final frequency samples, the overall storage requirement scales as $O(2P^2K)$. For instance, a 100-port structure with 10000 frequency samples requires more than 1.6 GB of storage using complex double-precision arithmetics.

This large storage requirement can be relaxed and significantly reduced with a modified scheduling approach that interleaves the application of $\mathcal{R}$ and $\mathcal{C}$ iterations. In fact, after each subband (ω_m, ω_{m+1}) is flagged after running the $\mathcal{C}$ check at both its endpoints, only the P eigenvalues along the diagonal of $\boldsymbol{\Lambda}_m$ need to be stored for the final identification of local singular value maxima. The idea is then, during the $\mathcal{R}$ refinement loop, to

- apply a $\mathcal{C}$ check whenever a triplet of adjacent samples $(\omega_{m-1}, \omega_m, \omega_{m+1})$ is finalized by the $\mathcal{R}$ check;
- flag subband (ω_m, ω_{m+1}) as soon as both samples are processed by a $\mathcal{C}$ check;
- free the memory from data that is not required by later $\mathcal{R}$ or $\mathcal{C}$ checks, and reuse it to store new samples data, as required by local refinement.

TABLE I
PEAK MEMORY USAGE DURING PARALLEL ADAPTIVE SAMPLING AND LOCAL PASSIVITY CHECK FOR A TEST CASE ($K = 4392, P = 56$) WITH (M_2) AND WITHOUT (M_1) MEMORY OPTIMIZATION. RESULTS ARE SHOWN FOR DIFFERENT NUMBER OF THREADS T.

T	M_1, MB	M_2, MB
1	442	21
2	446	24
3	451	28
4	455	34
5	461	32
6	471	39
7	480	41
8	491	50

The actual implementation does not free or allocate any memory during the main refinement loop, since this would dramatically impact performance (memory management operations require exclusive access to resources and are not thread-safe). We use a pool (buffer) of elementary memory cells that is preallocated, based on some heuristic criterion depending on the number of concurrent threads T. These cells are reused by suitable linking through pointer reassignment. If the preallocated memory pool is full, then another block is allocated at once, thus limiting impact on parallel performance. Table I illustrates the memory savings obtained for a significant test case. Note that this memory optimization is achieved with no loss of performance or parallel efficiency.

IV. PARALLEL HAMILTONIAN EIGENSOLUTION

As pointed in Sec. II-B, an alternative and purely algebraic passivity check that does not require any frequency sampling is provided by the set of purely imaginary eigenvalues of the Hamiltonian matrix (11). In particular, denoting these eigenvalues as $\mu_k = \jmath\bar{\omega}_k$, it is well known that the frequencies $\bar{\omega}_k$ provide a partition of the imaginary axis into open subbands $\bar{\Omega}_k$ such that

$$\sigma_i(\jmath\omega) \neq 1 \quad \forall\omega \in \bar{\Omega}_k \quad \forall k\,, \tag{30}$$

so that the number of singular values σ_i exceeding the passivity threshold $\gamma = 1$ does not change within each $\bar{\Omega}_k$. The frequencies $\bar{\omega}_k$ provide the only crossing points between the singular value trajectories and the passivity threshold, see Fig. 1 for an illustration.

The fast determination of the eigenvalues μ_k was discussed in [24]. Instead of forming the full Hamiltonian matrix (11) and computing the whole set of its eigenvalues, from which the purely imaginary ones are then extracted, it is possible to search only a small region of the complex plane covering the imaginary axis. This is obtained by a shift-and-invert [51]-[55] approach combined with a rational Arnoldi process [50], which amounts to forming an orthogonal basis for the Krylov subspace

$$\operatorname{span}\left\{\boldsymbol{v}_1, (\mathcal{H} - \theta\mathcal{I})^{-1}\boldsymbol{v}_1, \ldots, (\mathcal{H} - \theta\mathcal{I})^{-d+1}\boldsymbol{v}_1\right\} \tag{31}$$

where $\boldsymbol{v}_1$ is a suitable random starting vector and $\theta \in \jmath\mathbb{R}$ is a purely imaginary shift. A projection of the Hamiltonian eigenvalue problem onto this space returns a smaller-size matrix, whose eigenvalues approximate the eigenvalues of $\mathcal{M}$ that are closest to θ. By a suitable thresholding to monitor convergence as the size of the Krylov subspace grows, see [24], one obtains both few eigenvalue estimates $\{\tilde{\lambda}_i\}$ close to θ and an associated radius ρ, which defines a circular region $\mathcal{C}_{\theta,\rho}$ centered at θ where a sufficiently accurate estimate of all included Hamiltonian eigenvalues is available. This *single-shift* Arnoldi iteration $\mathcal{S}$ can be formally described in functional form as

$$(\{\lambda_i\}, \rho) \leftarrow \mathcal{S}(\theta, \rho^0) \tag{32}$$

where the input parameters are the shift and some initial radius ρ^0 defining a tentative eigenvalue search region, and the output parameters are the complete set of eigenvalues $\{\lambda_i\}$ that are included in the disk $\mathcal{C}_{\theta,\rho}$, whose radius ρ is not known in advance but is determined during the process.

The entire imaginary axis can be searched by placing multiple shifts $\theta_k \in \jmath\mathbb{R}$, and by finding an estimate of few Hamiltonian eigenvalues closest to each shift through dedicated single-shift iterations. A standard bisection process, documented in [24], allows to cover the entire frequency band of interest. Then, the eigenvalue sets obtained from all shifts are collected, and all purely imaginary eigenvalues $\jmath\bar{\omega}_k$ are extracted. In order to determine whether the model is passive within the corresponding subbands $\bar{\Omega}_k$, it is sufficient to compute the singular values of the model at a single point within each band.

The above multishift Arnoldi process is another excellent candidate for parallelization, since the numerical processing required by each shift θ_k is independent on each other shift $\theta_{k'\neq k}$. However, there is a hidden interdependency since the number and location of new shifts to be processed depends on the results obtained on previously analyzed shifts. The algorithm described in [40] overcomes these difficulties by adopting a dynamic scheduling process that, given a number of available threads T, partitions the frequency band to be checked into disjoint subbands, which are processed independently by different threads. A high-level description of this *multi-shift* scheme is outlined below, with reference to Fig. 5.

1) Start by subdividing the frequency band of interest $[0, \Omega_{\max}]$ into κT equal subintervals, where $\kappa \geq 2$, and define the corresponding edges as tentative shifts θ_k; the condition $\kappa \geq 2$ ensures that there are more subbands than threads;
2) the first $T-1$ shifts together with the last (centered at the right edge of the frequency band) are assigned to the T computing threads, which perform independent single-shift iterations (step 1 in Fig. 5);
3) whenever a thread computes its task, a new single-shift iteration is started, by centering it at the first available shift waiting to be processed (e.g., thread $t = 1$ in step 2 of Fig. 5);
4) when all initial shifts have been processed and a computing thread becomes idle, a new shift is defined at the midpoint of the frequency band $(\theta_{\text{left}} + \rho_{\text{left}}, \theta_{\text{right}} - \rho_{\text{right}})$, where $\theta_{\text{left}}, \theta_{\text{right}}$ denote two adjacent shifts that have already been processed and $\rho_{\text{left}}, \rho_{\text{right}}$ are the corresponding convergence radii. See, e.g., the reallocation of thread $t = 2$ from step 8 to step 9 in Fig. 5. Note that the choice $\kappa \geq 2$ in the initialization step 1) guarantees that a pair of adjacent completed shifts is always available, except at termination;
5) as a further optimization, anytime a thread completes its single-shift iteration, the obtained convergence circle is checked and any shifts that might be included (which therefore do not need any processing) are removed from the processing queue;
6) the algorithm stops when the entire frequency band of interest $[0, \Omega_{\max}]$ is covered by the union of all convergence circles obtained through the iteration,

$$\jmath[0, \Omega_{\max}] \subset \mathcal{U} = \bigcup_k \mathcal{C}_{\theta_k, \rho_k} . \tag{33}$$

7) whenever all single-shift iterations are completed and estimates for all eigenvalues $\tilde{\lambda}_k \in \mathcal{U}$ are available, all these estimates are refined down to machine precision through another set of shift-and-invert iterations centered

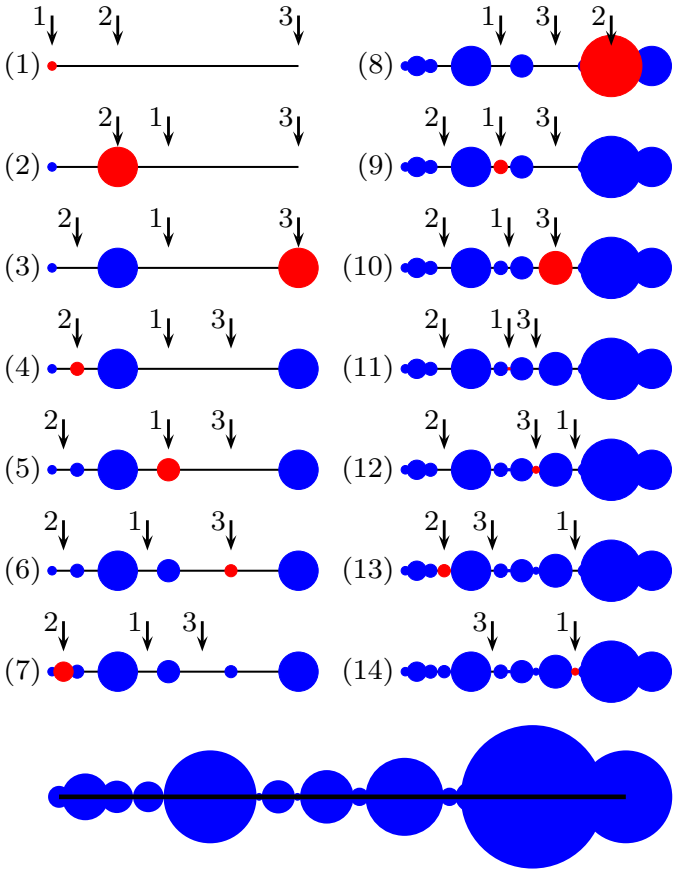

Fig. 5. Illustration of multishift iteration process using $T = 3$ threads. Each row represents a snapshot of the scanned frequency band at the end of each iteration, when one thread completes its $\mathcal{S}$ iteration and returns its convergence circle (depicted in red color). The arrows pinpoint the location of the shift θ under processing by the corresponding thread. Completed circles at previous iterations are represented with blue color. The situation at the end of the multishift iteration is depicted in the last row (enlarged view), showing that the union of all convergence circles covers the entire frequency band.

at $\tilde{\lambda}_k$. The parallelization of this last step is trivial, since a static scheduling is sufficient.

In summary, thread decoupling is achieved by the partitioning of step 1. Automatic load balancing is achieved by rescheduling idle threads at those frequency shifts where significant work is still required (steps 3 and 4). Finally, any unnecessary work is avoided by removing the corresponding shifts from the processing queue in step 5.

V. Combining adaptive sampling and Hamiltonian eigensolution

Both the adaptive sampling scheme of Sec. III and the Hamiltonian eigensolution of Sec. IV provide on output a subdivision of the frequency band $[0, \Omega_{\max}]$, in particular

$$\text{PAS} \quad \rightarrow \quad [0, \Omega_{\max}] = \left\{ \cup_q \Omega_q^{\text{p}} \right\} \bigcup \left\{ \cup_q \Omega_q^{\text{np},?} \right\} , \quad (34)$$

$$\text{Ham} \quad \rightarrow \quad [0, \Omega_{\max}] = \bigcup_k \bar{\Omega}_k \, , \quad (35)$$

where we have collected in the same set the non-passive and the undetermined bands from the adaptive sampling check. By construction, we know that

$$\forall \Omega_q^{\text{p}}, \quad \exists k : \; \Omega_q^{\text{p}} \subseteq \bar{\Omega}_k \, , \quad (36)$$

since the model is locally passive (with smooth tracking) at all computed samples of Ω_q^{p}.

The adaptive sampling process is very fast, but it does not provide a precise localization of all crossing points $\bar{\omega}_k$ of singular value trajectories with the passivity threshold. Moreover, the corresponding local passivity check is unable to qualify some frequency bands, which are still left undetermined. Conversely, the Hamiltonian eigensolution provides a precise characterization of each individual subband, but this higher resolution comes with a higher computational cost. This section describes our approach for combining the advantages of both schemes, in order to obtain the most precise information as possible with good parallel efficiency.

The basic idea is to exploit (36) in order to exclude the frequency bands Ω_q^{p} obtained through adaptive sampling from the more expensive search of Hamiltonian eigenvalues. Therefore, we perform first a parallel adaptive sampling and local passivity check to flag these certainly passive subbands, which are then removed from the starting interval

$$[0, \Omega_{\max}] \setminus \left\{ \cup_q \Omega_q^{\text{p}} \right\} = \cup_q \Omega_q^{\text{np},?} \, . \quad (37)$$

Only the remaining bands are searched for imaginary Hamiltonian eigenvalues. This is achieved by applying a set of local multishift iterations independently to each of the individual subbands $\Omega_q^{\text{np},?} = [\omega_q', \omega_q'']$. The same scheduling approach already described in Sec. IV is used, with some additional higher-level scheduling that manages the concurrent processing of multiple subbands.

Assuming T available threads and Q distinct subbands to be processed, our proposed dynamic scheduling is based on the following rules

1) the subbands to be characterized are sorted in decreasing order according to their bandwidth $\omega_q'' - \omega_q'$ and ranked; this ordering is maintained until the multishift iteration loop is completed;
2) if $Q \geq T$, a single thread is initially allocated to each band, starting from the top-ranked ones;
3) if instead $Q < T$, more than one thread is assigned to each subband. In the initialization stage, an even subdivision of threads among the subbands is used, with $\lceil T/Q \rceil$ and $\lfloor T/Q \rfloor$ threads assigned to the first and last subbands in the current ranking;
4) an individual multishift iteration, as in Sec. IV, is applied to each subband with its private set of threads, until this subband is completed and all its threads become idle;
5) when a number of threads become available, they are assigned to the frequency bands that are still under processing starting from the top-ranked one, following the same "democratic" strategy of item 3) above, unless there are some bands that are still waiting for the first thread. The latter are served first to improve parallel efficiency.

We see that, as soon as one subband is completed, its threads start helping the largest subbands still under processing, thus speeding up their completion with automatic load balancing. Therefore, the fine-grained subdivision of all threads among elementary operations provide an excellent opportunity for maintaining scalability and parallel efficiency when increasing the number of cores.

VI. Passivity enforcement

Several algorithms are available for enforcing model passivity, see [15]-[40] and references therein. Most of these schemes share the common strategy of applying some perturbation to the model coefficients so that the model becomes passive. This perturbation is invariably complemented by a suitable control over the perturbation amount, usually expressed in terms of some transfer matrix norm. Here, we follow the standard approach by applying the perturbation to the state matrix $\boldsymbol{C}$, which usually stores the residues matrices $\boldsymbol{R}_j$ in (4), as $\boldsymbol{C} \rightarrow \boldsymbol{C} + \boldsymbol{\Delta}$, and we minimize the energy of the corresponding input-output perturbation, expressed by

$$\mathcal{E} = \text{tr}\{\boldsymbol{\Delta} \boldsymbol{P} \boldsymbol{\Delta}^{\mathsf{T}}\} \, , \quad (38)$$

where tr denotes the matrix trace and $\boldsymbol{P}$ is the controllability Gramian [41] associated to the model state-space realization (2). Wherever appropriate, frequency-weighted versions of this perturbation norm can be used, by replacing $\boldsymbol{P}$ with a more general frequency-weighted Gramian matrix [31], [32].

The passivity constraints are here formulated as in (7) using the information that is collected from the previously completed passivity check. In particular, since the proposed passivity check scheme returns all local maxima $\hat{\sigma}_k$ of the singular value trajectories that exceed the threshold $\gamma = 1$, together with the frequencies $\hat{\omega}_k$ at which this maxima are attained, we set up multiple local passivity constraints at these frequencies, by relating the decision variables $\boldsymbol{\Delta}$ to the induced singular value perturbation $\delta\hat{\sigma}_k$ through a simple first-order approximation. The theory in [34] shows that the

resulting optimization problem can be formulated as

$$\min \mathrm{tr}\{\boldsymbol{\Delta P \Delta}^{\mathsf{T}}\} \quad \text{subject to} \quad \boldsymbol{W}\mathrm{vec}\{\boldsymbol{\Delta}\} \geq \boldsymbol{b}\,, \qquad (39)$$

where the vec operator stacks the columns of its matrix argument in a single column vector, and where the individual rows of matrix $\boldsymbol{W}$ and vector $\boldsymbol{b}$ correspond to the passivity constraint (7) applied to the perturbation of each local maximum $\hat{\sigma}_k$ at frequency $\hat{\omega}_k$. This formulation is standard.

Once (39) is solved, the model is checked again for passivity. In fact, the solution of the convex optimization (39) is not guaranteed to provide a passive model, since (39) provides a first-order approximation of the passivity constraint (7), which is formulated at a discrete subset of frequencies and not over the entire continuous imaginary axis. Therefore, the process must be iterated until the model results passive.

Being iterative in nature, the only possibility to parallelize the above passivity enforcement algorithm is to focus on each individual pass of the loop. The passivity check part, which actually dominates the computational cost, has been addressed in earlier sections. For what concerns the actual system perturbation, there are two main opportunities for parallelization, namely the construction of the local optimization problem (39) and its numerical solution. Our implementation is based on the following steps.

- The controllability Gramian $\boldsymbol{P}$ never changes through iterations. Therefore, $\boldsymbol{P}$ can be computed once (if needed) after the first passivity check. Since we are adopting a block-quasi-diagonal state-space realization, the structure of $\boldsymbol{P}$ is also block-diagonal, see [24]. Each diagonal block of this Gramian is therefore computed by an independent thread.
- The various rows of the passivity constraint in system (39) are independent, since based on different frequencies $\hat{\omega}_k$ and/or singular value maxima $\hat{\sigma}_k$. All these constraints are therefore spread among the available threads and computed in parallel. We remark that this computation requires the re-computation of the singular vectors $\boldsymbol{V}_k$ of the system transfer matrix, since our implementation does not keep track of these vectors to reduce the memory footprint. Also this computation is performed in parallel by all available computing threads.
- The solution of the constrained optimization problem (39) also requires an inner loop of iterations. Our implementation is based on an interior-point scheme [49], [59], [60], which embeds the set of inequality constraints (39) in the optimization cost function through a logarithmic barrier, and solves the resulting unconstrained optimization scheme through a globalized Newton iteration. Our parallelization intervenes in the construction of the linear system to be solved at each Newton iteration, which is performed concurrently by all available threads. For more details on this formulation, see [59], [61], [62],

In general, due to the several synchronization points that are required by this double iteration (with one outer loop constructing (39) after the passivity check, and one inner Newton loop for the actual solution of (39)), the overall parallel efficiency of the system perturbation alone is expected to be less than what can be achieved for the passivity check stage. Fortunately, this fact has little impact on the overall efficiency, since the computational cost of the passivity check is usually much larger than the cost of the perturbation stage. This fact will be confirmed by the numerical results presented in next Section.

TABLE II
TEST CASES: K, P AND N DENOTE THE NUMBER OF FREQUENCY SAMPLES IN THE RAW DATA, THE NUMBER OF PORTS AND THE DYNAMIC ORDER OF THE OBTAINED MODEL, RESPECTIVELY.

Case	K	P	N
1	71	92	1472
2	511	18	4572
3	4096	36	4968
4	2000	36	8064
5	570	34	1428
6	2043	18	2952
7	4096	18	3600
8	145	35	700
9	990	155	10540
10	282	164	6888
11	348	172	5504

VII. NUMERICAL RESULTS

Our proposed parallel passive macromodeling flow is illustrated on several benchmark cases, whose main features are summarized in Table II. The table shows that the number of ports P ranges from a minimum of 18 to a maximum of 172, and the dynamic order N (the size of the state matrix $\boldsymbol{A}$) from a minimum of 700 to a maximum of 10540. So, we can consider these benchmarks as ranging from medium-scale to large-scale models. All the underlying structures are electrical interconnects, in particular: cases 2–4, 6 and 7 are high-speed channels connecting CPU and memory in enterprise servers; cases 1 and 5 are package-level mixed signal/power distribution networks, case 8 is an interconnect model in a mixed-signal system, and cases 9–11 are chip-package-board power distribution models.

We analyze separately the performance of model generation via Parallel Vector Fitting in Sec. VII-A, the model passivity check via Parallel Adaptive Sampling, Hamiltonian eigenvalue computation, and by the proposed combination of both approaches in Sec. VII-B, and the global passivity enforcement in Sec. VII-C. All these results are presented by reporting the time required by a serial implementation, denoted in the following as τ_1, taken as the reference for measuring speedup and parallel efficiency, and the time required by our parallel implementation using 8 cores (τ_8) and 16 cores (τ_{16}). All tests were performed using a Linux server with four quad-core AMD Opteron processors running at 1.9 GHz. All algorithms were implemented in C/C++ based on the OpenMP paradigm [63] and the Lapack numerical libraries [64].

A. Model generation via Parallel Vector Fitting

We report here the timing results required by model generation using the PVF algorithm [14]. The original P-port scattering parameter data were subject to three PVF

TABLE III
TIMING RESULTS FOR QR FACTORIZATION OF THE LEAST SQUARES POLE RELOCATION SYSTEM IN THE PVF SCHEME. THE TIME REQUIRED BY 8 AND 16 THREADS IS REPORTED IN THE LAST TWO COLUMNS, TOGETHER WITH THE SPEEDUP FACTOR (IN BRACKETS) WITH RESPECT TO THE SERIAL IMPLEMENTATION USING 1 THREAD (SECOND COLUMN).

Case	τ_1, s	τ_8, s	τ_{16}, s
1	6.74	0.89 (7.59×)	0.46 (14.76×)
2	444.01	59.02 (7.52×)	30.90 (14.37×)
3	4088.17	533.90 (7.66×)	277.28 (14.74×)
4	104.72	13.89 (7.54×)	7.35 (14.25×)
5	373.64	49.65 (7.53×)	25.66 (14.56×)
6	43.10	5.83 (7.40×)	3.27 (13.20×)
7	628.17	82.56 (7.61×)	41.82 (15.02×)
8	1.39	0.19 (7.15×)	0.11 (13.21×)
9	601.55	77.83 (7.73×)	40.54 (14.84×)
10	569.44	72.77 (7.83×)	36.66 (15.53×)
11	374.20	48.10 (7.78×)	24.05 (15.56×)

TABLE IV
AS IN TABLE III, BUT REPORTING OVERALL TIME REQUIRED BY PVF TO GENERATE MACROMODELS FROM SCATTERING PARAMETER SAMPLES.

Case	τ_1, s	τ_8, s	τ_{16}, s
1	8.46	1.26 (6.70×)	0.73 (11.60×)
2	453.37	61.11 (7.42×)	32.41 (13.99×)
3	4114.51	539.81 (7.62×)	281.61 (14.61×)
4	105.74	14.11 (7.49×)	7.52 (14.06×)
5	433.79	63.05 (6.88×)	35.67 (12.16×)
6	47.44	6.80 (6.98×)	3.98 (11.92×)
7	633.88	83.86 (7.56×)	42.75 (14.83×)
8	1.48	0.21 (6.92×)	0.12 (12.40×)
9	615.82	81.04 (7.60×)	42.91 (14.35×)
10	650.62	90.53 (7.19×)	50.06 (13.00×)
11	405.33	54.70 (7.41×)	28.82 (14.06×)

pole relocation iterations by using a common pole set for all responses (using the response splitting scheme denoted as "none" in [14]), for which the most demanding part is a QR factorization stage of individual blocks of the large least-squares system required for pole relocation [11]. The parallelization of this algorithm part leads to the results of Table III, where also the speedup obtained by using $T = 8$ and $T = 16$ threads is reported within brackets. Table IV reports instead the overall time and corresponding parallel speedup factors required by the complete model generation via PVF. Although the parallel efficiency and the overall speedup is slightly less than observed for the QR factorization alone, we can conclude that an excellent performance is achieved by our parallel implementation, with close to ideal speedup factors for the large-scale cases that require significant computing time. These results confirm the PVF performance discussed in [14].

B. Passivity check

We now discuss the performance of the proposed passivity check schemes. The first set of results in Table V reports the number of frequency samples required by a continuous smooth tracking of the singular values/vectors. The set of initial samples $\mathcal{S}^0$ was generated using the guidelines of Sec. III-B, with $k_{\rm lin} = 300$ linearly spaced samples, $k_d = 4$ samples per decade over 9 decades of frequency, and $2R + 1 = 7$ samples per pole (in the common-pole model representation (4)). Since

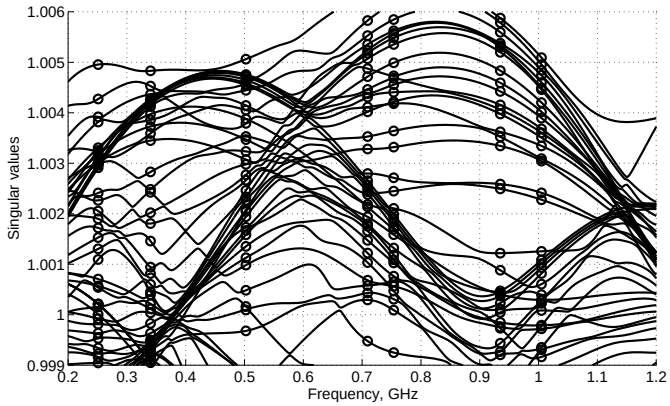

Fig. 6. Starting (circles) and final tracked frequency samples of few selected singular values for case 1

TABLE V
PASSIVITY CHECK: NUMBER OF INITIAL $\#\{\mathcal{S}^0\}$ AND FINAL $\#\{\mathcal{S}^{\rm end}\}$ FREQUENCY SAMPLES OBTAINED BY THE PROPOSED ADAPTIVE FREQUENCY SAMPLING SCHEME.

Case	$\#\{\mathcal{S}^0\}$	$\#\{\mathcal{S}^{\rm end}\}$
1	376	5229
2	1187	13216
3	766	6568
4	1093	16049
5	451	3129
6	873	6932
7	1007	10112
8	348	1969
9	558	12712
10	467	11229
11	429	10128

this number of initial samples is quite limited, it is expected that the PAS scheme will add many samples in order to track unambiguously the singular value trajectories. This is confirmed by the number of final samples $\#\{\mathcal{S}^{\rm end}\}$ reported in Table V, which is always in the order of several thousands. Figure 6 reports few selected singular value trajectories for case 1 within a restricted frequency band, showing how the final set of samples is able to resolve all fine variations of the curves, which are sampled too coarsely by the initial sample distribution.

Table VI reports the timing results and the parallel speedup for $T = 8$ and $T = 16$ concurrent threads obtained by our PAS scheme, inclusive of both adaptive sampling refinement and local passivity check. We see that the scalability of this passivity check scheme with the number of cores is excellent, with a speedup superior to $15\times$ in almost all cases.

We now turn to the Hamiltonian-based passivity check of Sec. IV. The timing results for all test cases are reported in Table VII, where the time required for the calculation of the Hamiltonian eigenvalues using a standard full eigensolver is also reported (second column) for comparison. We see that, even with a serial implementation, the required computation time by our multishift iterative solver is significantly reduced. This reduction becomes more aggressive if more threads are used, as demonstrated in Table VII. The speedup factors with

TABLE VI
TIMING RESULTS FOR THE PARALLEL ADAPTIVE SAMPLING AND LOCAL PASSIVITY CHECK SCHEME FOR $T = 1, 8$ AND 16 THREADS, WITH CORRESPONDING SPEEDUP FACTORS.

Case	τ_1, s	τ_8, s	τ_{16}, s
1	112.76	14.31 (7.88×)	7.11 (15.87×)
2	8.05	1.05 (7.6×)	0.51 (15.66×)
3	26.56	3.35 (7.93×)	1.85 (14.33×)
4	80.42	10.69 (7.52×)	5.09 (15.78×)
5	5.69	0.73 (7.80×)	0.37 (15.39×)
6	3.53	0.45 (7.89×)	0.23 (15.14×)
7	5.37	0.68 (7.87×)	0.35 (15.38×)
8	3.71	0.46 (8.00×)	0.24 (15.47×)
9	1151.28	145.83 (7.89×)	78.33 (14.70×)
10	1069.71	134.12 (7.98×)	69.46 (15.40×)
11	1091.38	136.44 (8.00×)	69.13 (15.79×)

TABLE VII
PASSIVITY CHECK VIA HAMILTONIAN EIGENVALUE COMPUTATION. THE TIME τ_T REQUIRED USING T THREADS IS REPORTED AND COMPARED TO THE TIME τ_{full} REQUIRED BY A FULL EIGENSOLVER.

Case	τ_{full}, s	τ_1, s	τ_8, s	τ_{16}, s
1	153.14	66.40	9.56 (6.94×)	5.70 (11.66×)
2	4303.77	358.81	49.73 (7.22×)	30.66 (11.70×)
3	5563.02	283.67	40.45 (7.01×)	22.28 (12.73×)
4	22288 (*)	929.37	131.14 (7.09×)	68.05 (13.66×)
5	150.11	34.77	4.54 (7.65×)	3.02 (11.51×)
6	1216.43	225.98	30.81 (7.34×)	19.42 (11.64×)
7	2239.23	264.78	34.75 (7.62×)	19.69 (13.45×)
8	18.58	23.73	3.18 (7.47×)	1.73 (13.72×)
9	48234 (*)	1718.56	268.54 (6.40×)	148.18 (11.60×)
10	14177 (*)	2594.00	361.90 (7.17×)	204.01 (12.72×)
11	7460 (*)	1711.48	246.21 (6.95×)	135.73 (12.61×)

(*) estimated based on theoretical $O(N^3)$ scaling law

$T = 8$ and $T = 16$ threads are a bit reduced with respect to the adaptive sampling check, but still quite satisfactory and always superior to $11\times$ with $T = 16$ threads, with a best case reaching $13.66\times$. This reduction in parallel efficiency is due to the higher degree of interdependency between concurrent threads and, especially, in the smaller amount of independent calculations (frequency shifts) available for thread distribution. For reference, in the PAS scheme for case 9 one has a number of final frequency samples of 12712, whereas only 326 frequency shifts are required by the selective Hamiltonian eigenvalue calculation. Also, the overall CPU time is higher for the Hamiltonian-based check with respect to the adaptive sampling check for most (but not all) cases. Despite this reduction in parallel efficiency and increased runtime, the Hamiltonian-based check allows a precise determination of the frequencies $\bar{\omega}_k$ that bracket the passivity violation bands, for which only an estimate is available from the adaptive sampling check.

The performance of the passivity check proposed in Sec. V based on the combination of adaptive sampling and multi-band Hamiltonian eigenvalue calculation is illustrated in Table VIII. This check, although more computationally expensive than the sampling-based check, provides a precise localization of passivity violations and is generally less expensive than the Hamiltonian-based check alone. Only few cases require a larger runtime (cases 1, 8, 9–11). This is due to the fact that

TABLE VIII
TIMING RESULTS FOR THE HYBRID PASSIVITY CHECK BASED ON ADAPTIVE SAMPLING COMBINED WITH SELECTED HAMILTONIAN EIGENVALUES.

Case	τ_1, s	τ_8, s	τ_{16}, s
1	171.97	22.20 (7.75×)	12.74 (13.50×)
2	9.53	1.57 (6.06×)	1.22 (7.82×)
3	28.24	4.22 (6.69×)	3.29 (8.58×)
4	110.89	17.03 (6.51×)	12.13 (9.14×)
5	28.46	3.90 (7.30×)	2.50 (11.40×)
6	3.44	0.44 (7.85×)	0.24 (14.61×)
7	5.55	0.70 (7.94×)	0.36 (15.37×)
8	28.50	3.60 (7.92×)	2.01 (14.20×)
9	2831.43	402.08 (7.04×)	218.95 (12.93×)
10	2957.15	419.10 (7.06×)	238.60 (12.39×)
11	2837.98	397.00 (7.15×)	221.38 (12.82×)

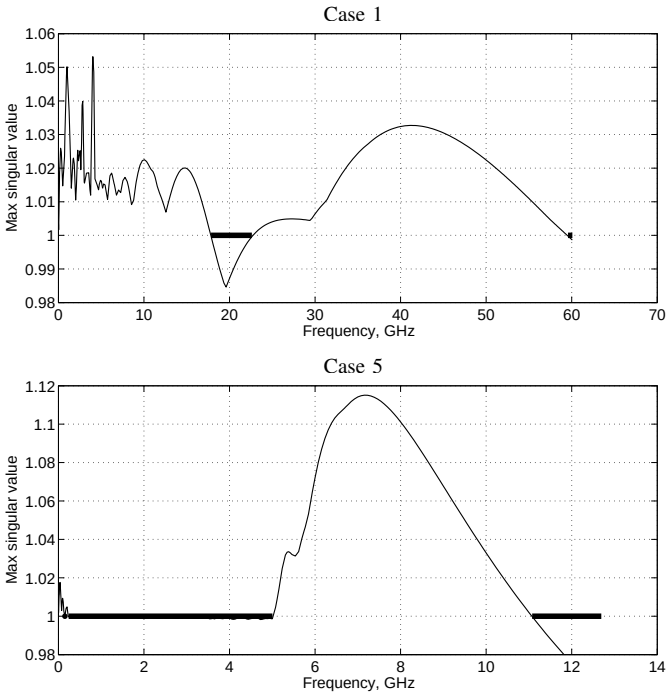

Fig. 7. Maximum singular value (thin line) and frequency bands Ω_q^{p} that are flagged as passive after the adaptive sampling check (thick line). Top panel: case 1; bottom panel: case 5.

the passivity violations for those cases are spread over a large frequency band rather than localized at small-size independent subbands. This implies that the set $\cup_q \Omega_q^{\text{p}}$ in (37) is small or even empty, so that a large portion or even the full frequency band $[0, \Omega_{\max}]$ has to be scanned again by the Hamiltonian eigensolver. The preliminary adaptive sampling leads to no advantage in such cases. A different scenario happens, e.g., for case 5, for which the set $\cup_q \Omega_q^{\text{p}}$ is a significant portion of $[0, \Omega_{\max}]$, so that only a small remaining subband necessitates Hamiltonian eigenvalue determination. A graphical illustration of these two scenarios for cases 1 and 5 is provided in Fig. 7.

C. Passivity enforcement and overall results

The timing results for the passivity enforcement step described in Sec. VI are reported in Table IX. From these results we can observe that the serial runtime is almost negligible in

TABLE IX
TIMING RESULTS AND PARALLEL SPEEDUP FOR THE PASSIVITY ENFORCEMENT.

Case	τ_1, s	τ_8, s	τ_{16}, s	Iterations
1	41.66	25.49(1.63×)	12.31 (3.38×)	8
2	8.47	5.64(1.50×)	0.40 (2.74×)	42
3	6.01	4.62(1.30×)	2.88 (2.08×)	18
4	5.88	4.08(1.44×)	2.78 (2.11×)	19
5	2.66	2.14(1.24×)	1.98 (1.34×)	29
6	2.13	1.98(1.08×)	1.95 (1.09×)	45
7	1.99	1.83(1.08×)	1.85 (1.07×)	38
8	0.98	0.88(1.11×)	1.00 (0.98×)	16
9	28.28	12.13(2.33×)	9.25 (3.05×)	10
10	112.34	39.51(2.84×)	26.72 (4.20×)	7
11	25.34	10.65(2.37×)	8.37 (3.03×)	33

TABLE X
TIMING RESULTS AND PARALLEL SPEEDUP FOR THE COMPLETE PASSIVITY ENFORCEMENT LOOP.

Case	τ_1, s	τ_8, s	τ_{16}, s
1	1250.94	182.47 (6.86×)	113.31 (11.04×)
2	804.99	121.58 (6.62×)	101.57 (7.93×)
3	675.65	87.73 (7.70×)	70.22 (9.62×)
4	2320.52	369.85 (6.27×)	340.35 (6.82×)
5	382.74	72.82 (5.26×)	62.63 (6.11×)
6	468.39	65.94 (7.10×)	55.90 (8.38×)
7	558.96	84.26 (6.63×)	51.49 (10.86×)
8	101.59	21.97 (4.63×)	16.22 (6.26×)
9	11992.74	1795.32 (6.68×)	1523.24 (7.87×)
10	7397.36	1279.82 (5.78×)	995.71 (7.43×)
11	30606.63	4520.92 (6.77×)	3223.43 (9.50×)

practically all cases with respect to the time required by the passivity check. This confirms that most attention in algorithm speedup via parallelization must be devoted to the passivity check phase. We also see that the parallel efficiency of the enforcement process is much worse than what was achieved for the passivity check. This is easily explained noting that passivity enforcement is achieved through an inner iteration loop for solving problem (39) based on an interior-point scheme [59], [61], [62]. The number of iterations for each case is reported in the last column of Table IX. Each iteration requires the construction and the solution of a relatively small-size unconstrained optimization problem, whose formulation is based on full dense matrices. The cost for the individual iteration is thus not expected to scale well with the number of concurrent threads used to solve this problem. Moreover, when the individual cost per iteration is small, no advantage at all is expected from parallelization. Some moderate speedup is observed for those cases that require significant runtime per iteration.

We now turn to the complete passivity enforcement loop, which iteratively performs a passivity check and perturbs the model coefficients by solving (39) until the model is passive. In order to reduce overall runtime, the passivity check is performed in our implementation only by the PAS scheme, until no passivity violations are detected. At this point, the hybrid check of Sec. V is used in order to precisely detect and eliminate the residual passivity violation bands.

The timing results for all cases are reported in Table X. We see that, despite the several unavoidable synchronization points

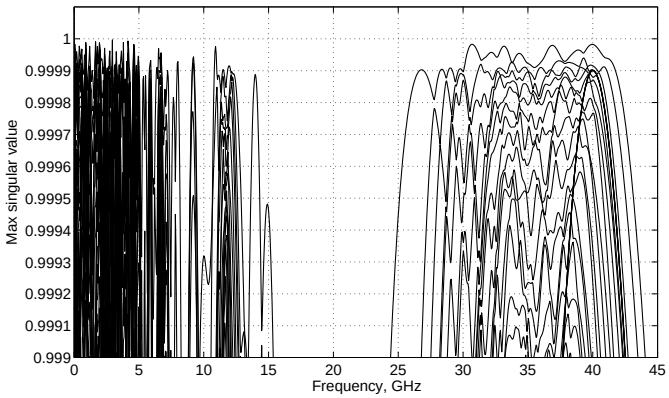

Fig. 8. Trajectories of all singular values for case 1 model after passivity enforcement (vertical scale has been stretched around the passivity threshold $\sigma = 1$).

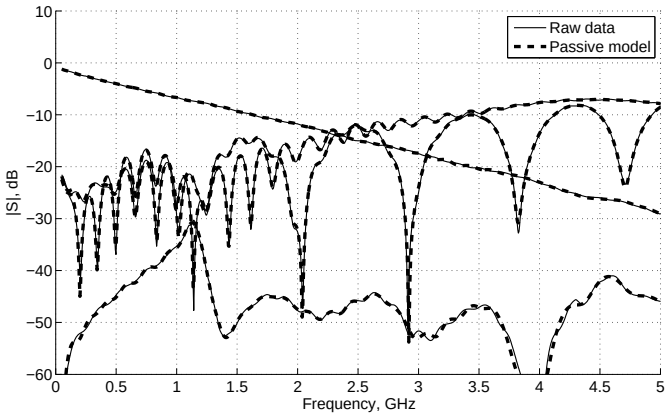

Fig. 9. Comparison between passive model and raw data for some case 3 responses. Similar results (not shown) were obtained for all other responses and all other cases.

possibly affecting the iterative scheme, the overall speedup obtained by using $T = 8$ and $T = 16$ threads is quite satisfactory. As an example, we report in Fig. 8 the set of singular values of the case 1 model after passivity enforcement. The enlarged vertical scale shows that all singular values are bounded by one, implying model passivity. Figure 9 compares a selected set of scattering responses of the final passive case 1 model to the original data used for model extraction. The accuracy is excellent, with no visual difference between model and data on this scale.

We conclude this section by reporting in Fig. 10 the speedup factor τ_T/τ_1 achieved by the four key stages of our proposed parallel macromodeling flow, namely the PVF model extraction, the passivity check via adaptive sampling, the hybrid adaptive sampling/Hamiltonian passivity check, and the full passivity enforcement loop. The plots report best case, worst case, and average among all analyzed benchmarks. These plots clearly illustrate the benefits of parallelization.

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

We have presented a comprehensive macromodel extraction flow based on a parallel formulation and implementation of

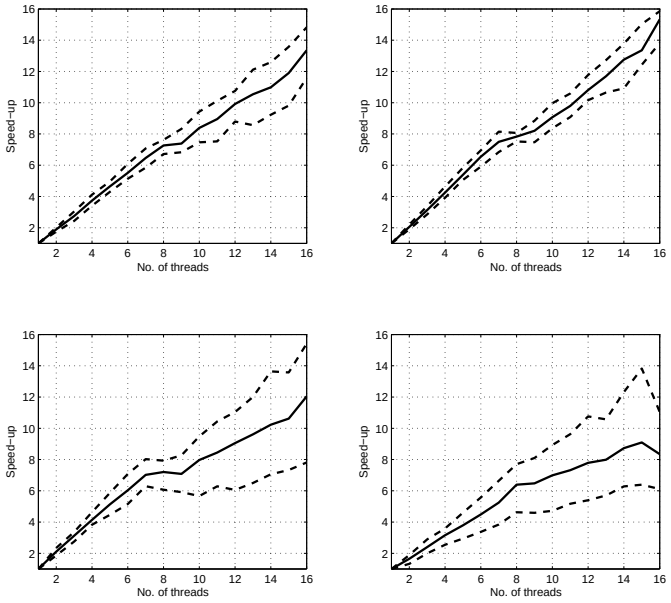

Fig. 10. Speedup plots for the four stages of our parallel macromodeling flow: PVF (top left), PAS passivity check (top right), hybrid PAS/Hamiltonian passivity check (bottom left), and complete passivity enforcement (bottom right). The plots report the ratio τ_T/τ_1 versus the number T of computational threads for the best and worst cases (dashed lines), and the average (solid lines) among all analyzed benchmarks.

all underlying numerical algorithms. The starting point is a set of tabulated frequency responses, and the final objective is a macromodel in state-space form, whose transfer function approximates with good accuracy the original data. The standard flow is applied here, based on rational curve fitting followed by a passivity enforcement step via model perturbation.

The main focus of this work is the acceleration of all steps of the macromodel extraction flow through algorithm parallelization and deployment on multicore computing architectures. To this end, we reformulated both rational curve fitting, passivity check and enforcement so that the various numerical operations can be performed concurrently by independent computing threads assigned to the various available processors. Performance and scalability tests were performed by increasing the number of threads up to the largest allowed by our server (16).

The numerical results show that macromodel extraction for medium and large scale structures can be performed fast and efficiently. The most demanding part remains passivity enforcement, which requires a passivity check at each iteration. Good scalability with the number of computing threads is observed at all stages of the extraction, providing a quite promising framework for deployment on future computing architectures, which are expected to offer massively parallel computing power even at the desktop level.

Various directions are available for future investigations. We mention the possibility to extend the proposed parallel macromodeling flow to electrically large structures by embedding delay terms in the macromodels. This subject has been studied in several publications, see e.g. [65]-[69] and references therein. It is expected that an ad hoc parallelization strategy will be quite effective in reducing model extraction also in this case.

A second direction is pointed by [39], where a new framework for passivity enforcement based on a convex nonsmooth formulation was developed. This approach provides a theoretical proof of optimality and convergence, at the price of a much increased iteration count, hence runtime, with respect to the approach of this work. Each iteration of [39] still requires a full passivity check; it is therefore expected that very similar speedup factors as documented in this work will be possible through parallelization. However, we believe that research efforts within this framework should be directed first to the reduction in the number of required iterations, leaving code optimization and parallelization to a second stage.

REFERENCES

[1] M. Nakhla and R. Achar, "Simulation of High-Speed Interconnects", *Proc. IEEE*, Vol. 89, No. 5, pp. 693–728, May 2001.

[2] M. Celik, L. Pileggi, A. Obadasioglu, *IC Interconnect Analysis*, Kluwer, 2002.

[3] W.Beyene, J. Schutt-Ainé, "Accurate Frequency-Domain Modeling and Efficient Circuit Simulation of High-Speed Packaging Interconnects", *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1941-1947, Oct. 1997

[4] K.L.Choi, M.Swaminathan, "Development of Model Libraries for Embedded Passives Using Network synthesis", *IEEE Trans. Circuits and Systems II*. vol. 47, pp. 249–260, Apr. 2000.

[5] M. Elzinga, K. Virga, L. Zhao, J.L.Prince, "Pole-Residue Formulation for transient simulation of high-frequency interconnects using Householder LS curve-fitting techniques", *IEEE Trans. Comp. Packag. Manufact. Technol.*, vol. 23, pp. 142–147, Mar. 2000

[6] M. Elzinga, K. Virga, J.L.Prince, "Improve Global Rational Approximation Macromodeling Algorithm for Networks Characterized by Frequency-Sampled Data", *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 1461–1467, Sept. 2000

[7] J. Morsey, A. C. Cangellaris, "PRIME: Passive Realization of Interconnects Models from Measures Data", in *Proc. IEEE* 10^{th} *Topical Meeting on Electr. Perf. of Electron. Packag.*, 2001, pp. 47–50.

[8] B. Gustavsen, A. Semlyen, "Rational approximation of frequency responses by vector fitting", *IEEE Trans. Power Delivery*, Vol. 14, N. 3, pp. 1052–1061, July 1999.

[9] B. Gustavsen, A. Semlyen, "A robust approach for system identification in the frequency domain", *IEEE Trans. Power Delivery*, Vol. 19, N. 3, pp. 1167–1173, July 2004.

[10] D. Deschrijver, B. Haegeman, T. Dhaene, "Orthonormal Vector Fitting: A Robust Macromodeling Tool for Rational Approximation of Frequency Domain Responses", *IEEE Transactions on Advanced Packaging*, Vol. 30, No. 2, pp. 216–225, May 2007.

[11] D. Deschrijver, M. Mrozowski, T. Dhaene, D. De Zutter, "Macromodeling of Multiport Systems Using a Fast Implementation of the Vector Fitting Method," *IEEE Microwave and Wireless Components Letters*, Vol. 18, N. 6, June 2008, pp.383–385.

[12] S. Grivet-Talocia, M. Bandinu, "Improving the Convergence of Vector Fitting in Presence of Noise", *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, n. 1, pp. 104-120, February, 2006.

[13] S. B. Olivadese, S. Grivet-Talocia, "Compressed Passive Macromodeling" , *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, Vol. 2, n. 8, pp. 1378-1388, August, 2012.

[14] A. Chinea and S. Grivet-Talocia, "On the parallelization of vector fitting algorithms," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, pp. 1761–1773, November 2011.

[15] B. Gustavsen, A. Semlyen, "Enforcing passivity for admittance matrices approximated by rational functions", *IEEE Trans. Power Systems*, Vol. 16, 2001, 97–104.

[16] B. Gustavsen, "Computer Code for Passivity Enforcement of Rational Macromodels by Residue Perturbation," *IEEE Trans. Adv. Packaging*, vol. 30, No. 2, pp. 209–215, May 2007.

[17] C. P. Coelho, J. Phillips, L. M. Silveira, "A Convex Programming Approach for Generating Guaranteed Passive Approximations to Tabulated Frequency-Data", *IEEE Trans. Computed-Aided Design of Integrated Circuits and Systems*, Vol. 23, No. 2, pp. 293–301, Feb. 2004.

[18] H. Chen, J. Fang, "Enforcing Bounded Realness of S parameter through trace parameterization", in *12th IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, October 27–29, 2003, Princeton, NJ, pp. 291–294.

[19] S. Grivet-Talocia, "Enforcing Passivity of Macromodels via Spectral Perturbation of Hamiltonian Matrices," in *7th IEEE Workshop on Signal Propagation on Interconnects, Siena, Italy*, May 11–14, 2003, pp. 33–36.

[20] D. Saraswat, R. Achar, M. Nakhla, "Enforcing Passivity for Rational Function Based Macromodels of Tabulated Data", in *12th IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, October 27–29, 2003, Princeton, NJ, pp. 295–298.

[21] D. Saraswat, R. Achar and M. Nakhla, "A Fast Algorithm and Practical Considerations For Passive Macromodeling Of Measured/Simulated Data", *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Vol. 27, pp. 57–70, Feb. 2004.

[22] D. Saraswat, R. Achar and M. Nakhla, "Global Passivity Enforcement Algorithm for Macromodels of Interconnect Subnetworks Characterized by Tabulated Data", *IEEE Transactions on VLSI Systems*, Vol. 13, No. 7, pp. 819–832, July 2005.

[23] S. Grivet-Talocia, "Passivity enforcement via perturbation of Hamiltonian matrices", *IEEE Trans. CAS-I*, pp. 1755-1769, vol. 51, n. 9, September, 2004

[24] S. Grivet-Talocia, A. Ubolli "On the Generation of Large Passive Macromodels for Complex Interconnect Structures", *IEEE Trans. Adv. Packaging*, vol. 29, No. 1, pp. 39–54, Feb. 2006

[25] S. Grivet-Talocia, "Improving the efficiency of passivity compensation schemes via adaptive sampling", *14th IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, Austin, Texas (USA), October 24–26, 2005, pp. 231–234.

[26] A. Semlyen, B. Gustavsen, "A half-size singularity test matrix for fast and reliable passivity assessment of rational models", *IEEE Trans. Power Delivery*, Vol. 24, No. 1, pp. 345–351, Jan. 2009.

[27] B. Gustavsen, A. Semlyen, "Fast passivity assessment for S-parameter rational models via a half-size test matrix", *IEEE Trans. Microwave Theory and Techniques*, Vol. 56, No. 12, pp. 2701–2708, Dec. 2008.

[28] S. Grivet-Talocia, "An adaptive sampling technique for passivity characterization and enforcement of large interconnect macromodels," *IEEE Trans. Adv. Packaging*, Vol. 30, No. 2, pp. 226–237, May 2007.

[29] B. Gustavsen, "Fast passivity enforcement of Rational Macromodels by Perturbation of Residue Matrix Eigenvalues", *11th IEEE Workshop on Signal Propagation on Interconnects, May 13–16, 2007, Ruta di Camogli, Genova, Italy*, pp. 71–74.

[30] A Lamecki and M. Mrozowski, "Equivalent SPICE Circuits With Guaranteed Passivity From Nonpassive Models," IEEE Transactions on Microwave Theory And Techniques, Vol. 55, No. 3, pp. 526–532, Mar. 2007.

[31] S. Grivet-Talocia, A. Ubolli, "Passivity Enforcement With Relative Error Control" , *IEEE Trans. Microwave Theory and Techniques*, Vol. 55, No. 11, pp. 2374–2383, Nov. 2007.

[32] A. Ubolli, S. Grivet-Talocia, "Weighting Strategies for Passivity Enforcement Schemes", *16th IEEE Topical Meeting on Electrical Performance of Electronic Packaging, Atlanta, GA, 29–31 October, 2007*

[33] C. S. Saunders, Jie Hu, C. E. Christoffersen, M. B. Steer, "Inverse Singular Value Method for Enforcing Passivity in Reduced-Order Models of Distributed Structures for Transient and Steady-State Simulation, " *IEEE Trans. Microwave Theory and Techniques*, Vol. 59, No. 4, pp. 837–847, Apr. 2011.

[34] S. Grivet-Talocia and A. Ubolli, "A comparative study of passivity enforcement schemes for linear lumped macromodels," *IEEE Trans. Advanced Packaging*, vol. 31, pp. 673–683, Nov 2008.

[35] Z. Ye, L. M. Silveira, and J. R. Phillips, "Fast and Reliable Passivity Assessment and Enforcement with Extended Hamiltonian Pencil," in *International Conference on Computer Aided Design*, 2009, pp. 774–778.

[36] Z. Ye, L. M. Silveira, and J. R. Phillips, "Extended Hamiltonian Pencil for Passivity Assessment and Enforcement for S-parameter Systems," in *DATE 2010 Conference*, pp. 1148–1152.

[37] Z. Zhang, C. U. Lei, and N. Wong, "GHM: A generalized Hamiltonian method for passivity test of impedance/admittance descriptor systems," in *Proc. Int. Conf. Comput.-Aided Design, San Jose, CA*, Nov. 2009, pp. 767–773.

[38] Z. Zhang and N. Wong, "Passivity test of immittance descriptor systems based on generalized Hamiltonian methods," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 57, no. 1, pp. 61–65, Jan. 2010.

[39] G. Calafiore, A. Chinea, S. Grivet-Talocia, "Subgradient Techniques for Passivity Enforcement of Linear Device and Interconnect Macromodels", *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, n. 10, pp. 2990-3003, October 2012.

[40] L. Gobbato, A. Chinea, S. Grivet-Talocia, "A Parallel Hamiltonian Eigensolver for Passivity Characterization and Enforcement of Large Interconnect Macromodels," *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2011*, Grenoble, France, March 14-18, 2011, pp.26–31.

[41] T. Kailath, *Linear systems*, Englewood Cliffs, NJ:Prentice Hall, 1980.

[42] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. Canavero, R. Achar, "Stability, Causality, and Passivity in Electrical Interconnect Models" , *IEEE Trans. Adv. Packaging*, Vol. 30, No. 4, pp. 795–808, Nov. 2007.

[43] S. Grivet-Talocia, "On driving non-passive macromodels to instability", *Int. J. Circuit Theory Appl.*, 2009, in press.

[44] M. R. Wohlers, *Lumped and Distributed Passive Networks*, New York: Academic Press, 1969.

[45] R. Achar, M. Nakhla, "Minimum realization of reduced-order high-speed interconnect macromodels", in *Signal Propagation on Interconnects*, H. Grabinski and P. Nordholz Eds., Kluwer, 1998.

[46] G. H. Golub, C. F. van Loan, *Matrix computations*, 3^{rd} ed., Baltimore: Johns Hopkins University Press, 1996

[47] B.D.O. Anderson, S. Vongpanitlerd, *Network Analysis and Synthesis*, New York: Dover, 2006.

[48] S. Boyd, V. Balakrishnan, P. Kabamba, "A bisection method for computing the H_∞ norm of a transfer matrix and related problems", *Math. Control Signals Systems*, Vol. 2, 1989, pp. 207–219.

[49] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, *Linear matrix inequalities in system and control theory*, *SIAM studies in applied mathematics*, Philadelphia: SIAM, 1994.

[50] W. E. Arnoldi, " The principle of minimized iterations in the solution of the matrix eigenvalue problem", *Quart. Appl. Math.*, vol. 9, pp. 17-29, 1951.

[51] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide.* Philadelphia:SIAM, 2000.

[52] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, New York: Halsted Press, 1992.

[53] Y. Saad, "Variations on Arnoldi's Method for Computing Eigenelements of Large Unsymmetric Matrices", *Linear Algebra and its Applications*, Vol. 34, 1980, pp.269–295.

[54] V. Mehrmann, D. Watkins, Structure-preserving methods for computing eigenpairs of large sparse Skew-Hamiltonian/Hamiltonian pencils", *SIAM J. Sci. Comput.*, Vol. 22, No. 6, 2001, pp. 1905–1925

[55] I. M. Elfadel and D. L. Ling, "A block rational Arnoldi algorithm for multipoint passive model order reduction of multiport RLC networks," in *Proc. Int. Conf. Computer Aided Design*, Nov. 1997, pp. 66–71.

[56] E. Chiprout, M.S.Nakhla, "Analysis of Interconnects Networks using Complex Frequency Hopping (CFH)", *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 186-200, Feb. 1995.

[57] J. H. Wilkinson, "The algebraic eigenvalue problem," Oxford University Press, 1965.

[58] Raines, B.D.; Rojas, R.G.; , "Wideband Characteristic Mode Tracking," *IEEE Transactions on Antennas and Propagation*, vol.60, no.7, pp.3537-3541, July 2012.

[59] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[60] S. Boyd, *Lecture notes and slides for EE364b, Convex Optimization II*, Stanford University.

[61] B.T. Polyak, *Introduction to Optimization*, Optimization Software, 1987.

[62] D.P. Bertsekas, A. Nedic, A.E. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, 2003.

[63] OpenMP Architecture Review Board, *OpenMP C and C++ Application Program Interface - Version 2.0*, Mar. 2002. Available: http://www.openmp.org/mp-documents/cspec20.pdf.

[64] *LAPACK Users' Guide - Third edition*, Aug. 1999, Available: http://www.netlib.org/lapack/lug

[65] A. Chinea, P. Triverio, S. Grivet-Talocia, "Delay-Based Macromodeling of Long Interconnects from Frequency-Domain Terminal Responses," *IEEE Transactions on Advanced Packaging*, Vol. 33, No. 1, pp. 246–256, Feb. 2010.

[66] P. Triverio, S. Grivet-Talocia, A. Chinea, "Identification of highly efficient delay-rational macromodels of long interconnects from tabulated frequency data," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 3, pp. 566–577, 2010.

[67] A. Charest, M. Nakhla, R. Achar, D. Saraswat, N. Soveiko, I. Erdin, "Time Domain Delay Extraction-Based Macromodeling Algorithm for Long-Delay Networks," *IEEE Transactions on Advanced Packaging*, Vol. 33, No. 1, pp. 219–235, Feb. 2010.

[68] Chen, C., Saraswat, D., Achar, R., Gad, E., Nakhla, M. and Yagoub, M.C.E., "Passivity Compensation Algorithm for Method of Characteristics-based Multiconductor Transmission Line Interconnect Macromodels, *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 17, no. 8, pp. 1061–1072, 2009.
[69] Chen, C., Gad, E., Nakhla, M. and Achar, R., "Passivity Verification in Delay-based Macromodels of Multiconductor Electrical Interconnects", *IEEE Transactions on Advanced Packaging*, Vol. 30, no. 2, pp.246–256, 2007

Alessandro Chinea received the Laurea Specialistica (M.Sc.) and Ph.D. degrees in electronic engineering from Politecnico di Torino, Italy in 2006 and 2010, respectively. In 2009 he spent a period at the Department of Information Technology (INTEC) of the Ghent University, Belgium, working under the supervision of Professors T. Dhaene and L. Knockaert. Since 2012 he works at the IdemWorks s.r.l. as a senior engineer. His research interests concern passive macromodeling of electrical interconnects for electromagnetic compatibility and signal/power integrity problems. Dr. Chinea received the Optime Award from the Unione Industriale di Torino and he was selected for the IBM EMEA Best Student Recognition Event 2006.

Stefano Grivet-Talocia (M'98–SM'07) received the Laurea and the Ph.D. degrees in electronic engineering from Politecnico di Torino, Italy. From 1994 to 1996, he was with the NASA/Goddard Space Flight Center, Greenbelt, MD, USA. Currently, he is an Associate Professor of Circuit Theory with Politecnico di Torino. His research interests are in passive macromodeling of lumped and distributed interconnect structures, modeling and simulation of fields, circuits, and their interaction, wavelets, time-frequency transforms, and their applications. He is author of more than 120 journal and conference papers. He is co-recipient of the 2007 Best Paper Award of the IEEE Trans. Advanced Packaging. He received the IBM Shared University Research (SUR) Award in 2007, 2008 and 2009. Dr. Grivet-Talocia served as Associate Editor for the IEEE TRANSACTIONS ON ELECTROMAGNETIC COMPATIBILITY from 1999 to 2001. He is co-founder and President of IdemWorks.

Salvatore Bernardo Olivadese received the Laurea degree (B.Sc) in Information Technology in 2007 and the Laurea Specialistica degree (M.Sc) in Electronic Engineering in 2009, both from Politecnico di Torino, Italy, where He is currently working towards his Ph.D. degree. For his bachelor thesis He developed a Web2.0 service for scientific parallel applications, and for his master thesis He spent 6 months as visiting researcher at the Interconnect and Packaging Analysis Group of IBM T.J. Watson Research Center, Yorktown, NY, developing an Adaptive Frequency Sampling method. He was recipient of the IBM PhD Fellowship Award for the academic year 2011/12. His research interests concern packaging, circuit theory, Signal and Power Integrity, and parallel algorithms. Part of this research is conducted in collaboration with IBM, Intel, and IdemWorks.

Luca Gobbato received the Laurea Specialistica (M.Sc.) in electronic engineering from Politecnico di Torino in 2010. In 2011 he spent a period at the Interuniversity Research Center on Enterprise Networks, Logistic and Transportation, working under the supervision of professor G. T. Crainic. Since 2012 he works towards his Ph.D. degree at Politecnico di Torino. His research interests concern parallel programming and operational research. Dr. Gobbato received the degree award from Confindustria Servizi Innovativi e Tecnologici and he was selected for the IBM EMEA Best Student Recognition Event 2010.