

Fast Passivity Enforcement for S -Parameter Models by Perturbation of Residue Matrix Eigenvalues

Bjørn Gustavsen, *Senior Member, IEEE*

Abstract—Rational macromodels must be passive in order to guarantee a stable simulation. This paper introduces a fast approach for enforcing passivity for S -parameter based pole-residue models, using a similar method previously introduced for Y -parameter models. The approach is based on perturbing the elements of the residue matrices with the perturbed residue matrix eigenvalues as free variables. This gives large savings for the CPU time and memory requirements. The implementation does not require sparse computations. Combining the passivity enforcement step with iterations and fast passivity assessment via a half-size test matrix gives a globally passive model. Error control strategies are implemented via least squares weighting. The approach is demonstrated for a two-port microwave filter, a four-port interconnect, a 48-port low-order package model, and a 28-port high-order package model.

Index Terms—Macromodel, passivity, passivity enforcement, perturbation, rational model.

I. INTRODUCTION

RELIABLE time-domain simulation of high-speed electronic systems requires that each part of the system is modeled with sufficient accuracy. For instance, wave reflection and frequency-dependent distortion effects must be included in simulation of interconnects [1]. Similarly, the design and optimization of electronic packages requires accurate representation of the structure behavior over a wide frequency band.

Linear components and systems are often modeled via rational macromodels that emulate the behavior of the component with respect to a set of ports. The macromodel can easily be included in a SPICE-like simulation environment via a lumped circuit representation [2] or recursive convolution [3]. The characterization of the port behavior can come from electromagnetic calculations or measurements in the frequency domain or the time domain. The actual modeling from tabulated port data can easily be done using the pole relocating method known as vector fitting [4]–[11]. Vector fitting produces a multiport

Manuscript received August 04, 2008; revised September 29, 2008. The work was supported in part by the Norwegian Research Council (PETROMAKS Programme), in part by Compagnie Deutsch, in part by FMC Technologies, in part by Framo, in part by Norsk Hydro, in part by Petrobras, in part by Siemens, in part by Statoil, in part by Total, and in part by Vetco Gray. This work was recommended for publication by Associate Editor J. Tan upon evaluation of the reviewers comments.

The author is with SINTEF Energy Research, N-7465 Trondheim, Norway (e-mail: bjorn.gustavsen@sintef.no).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TADVP.2008.2010508

common-pole model with guaranteed stable poles. A major difficulty is that the extracted model is often nonpassive, i.e., it generates energy under certain port conditions. This can lead to unstable simulations in an unpredictable manner. The stability problem can be avoided by enforcing passivity during the fitting process using convex optimization [12], but that approach often requires lengthy computation times [13].

A perturbation method for passivity enforcement was introduced in [14], [15] for use with admittance (Y -) parameter models. (A similar approach for scattering (S -) parameter models was adopted in [16]). Passivity is enforced at discrete frequencies by perturbing the model residues such that the change to the admittance matrix is minimal in a considered frequency band (in-band). For package applications, the computational efficiency is low due to the need for solving a large and sparse quadratic programming problem, although sparse solvers can greatly improve the efficiency [17]. In addition, iterations are in general needed which are prone to divergence. A very fast procedure for passivity enforcement was proposed in [18]. That method perturbs the model residues indirectly via the Hamiltonian matrix eigenvalues to give a minimal change to the model impulse energy. The resulting change to the model behavior is higher than with the approach in [14], since it cannot discern between in-band and out-of band passivity violations, and because it lacks least squares weighting capability. In addition, it often requires more iterations [13]. Recent work has alleviated some of the accuracy problem [19], but divergence can still take place. Perturbation of poles was introduced in [20], leading to a computationally efficient approach. The perturbation of poles effectively scales each pole-residue term by a common factor, thereby giving a larger change to the model behavior than with residue perturbation. In [21], the speed problem of the residue perturbation method [14] was overcome by choosing the residue matrix eigenvalues as free variables, avoiding the need for sparse solvers. When combined with passivity assessment via the Hamiltonian matrix eigenvalues and a robust iterative technique [17] for prevention of divergence, an efficient and reliable approach was achieved for use with admittance-based pole-residue models. Modal perturbation was introduced for Y -parameter models in [22], [21] in order to preserve the relative accuracy of the model's modes. That feature appears less useful for S -parameter models since the ratio between the largest and smallest eigenvalue of S is much smaller than with Y .

Precise localization of passivity violations was introduced in [23], [18], [15], based on the eigenvalues of a Hamiltonian matrix associated with the model. The computational efficiency has later been improved by usage of dedicated solvers [24] and

fast sweep methods [25], [26]. Half-size test matrices have been introduced for the use with symmetrical models in [27], [28] (Y -parameters), and [29] (S -parameters).

In this paper, perturbation of residue matrix eigenvalues is adopted for the use with symmetrical scattering parameter based models on pole-residue form. Although much of the applied methodology has been presented in the past [14], [21], [29], the current paper shows the complete procedure for use with scattering parameters (rather than admittance parameters [21]) and it includes many details related to the implementation. In particular, the building of the constraint matrix is shown in detail. In addition, the applicability of the procedure is demonstrated for cases with many ports. Passivity entails that all singular values of the scattering matrix \mathbf{S} are smaller than unity, at all frequencies. An augmented matrix is formed whose eigenvalues are equal to the singular values of \mathbf{S} . Eigenvalue perturbation theory is used for relating a perturbation of the residue matrices to the singular values. The constraint on singular values is applied at carefully selected frequencies and combined with least squares minimization of the change to \mathbf{S} over a given frequency band. The problem size is reduced by choosing the (perturbed) residue matrix eigenvalues as free variables, thereby saving memory and computation time. The building of the system matrices is shown in detail with consideration to efficient memory handling and the use of column scaling for improved numerical robustness. The approach is combined with iterations and fast passivity assessment by a half-size test matrix. The approach is demonstrated for a two-port microwave filter, a four-port interconnect, a 48-port low-order package model, and a 24-port high-order package model. Finally, the numerical performance is compared with an implementation using residue perturbation and a sparse solver.

II. PASSIVITY REQUIREMENT

This work considers passivity enforcement of a symmetrical n -port pole-residue model (1). The poles $\{a_m\}$ and residue matrices $\{\mathbf{R}_m\}$ are either real or come in complex conjugate pairs, and all $\{\mathbf{R}_m\}$ are symmetrical. The constant term \mathbf{D} is real and symmetrical, and possibly zero. This type of model can be obtained by subjecting the scattering matrix $\mathbf{S}_{\text{data}}(s)$ to rational approximation by vector fitting [4]. For the purpose of passivity assessment, we will convert the pole-residue model into a state-space model with parameters $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$. The conversion process amounts to simple rearrangement of parameters given in the pole-residue model. The details are shown in [29]

$$\begin{aligned} \mathbf{S}_{\text{data}}(s) &\cong \mathbf{S}(s) = \sum_{m=1}^N \frac{\mathbf{R}_m}{s - a_m} + \mathbf{D} \\ &= \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}. \end{aligned} \quad (1)$$

The model is passive if all singular values σ_i of \mathbf{S} are smaller than unity, at all frequencies $s = j\omega$ [30], [31]. Thus, for the singular value decomposition (2), condition (3) is to be satisfied, where Σ is a diagonal matrix that contains the singular values. The number of singular values, n , is equal to the number of ports, i.e., the dimension of \mathbf{S} . In (2), the superscript H denotes hermitian (transposed and conjugated)

$$\mathbf{S}(s) = \mathbf{U}(s)\Sigma(s)\mathbf{V}^H(s) \quad (2)$$

$$\sigma_i(s) < 1, \quad i = 1 \dots n. \quad (3)$$

III. PASSIVITY ASSESSMENT

A. Identifying Bands of Passivity Violations

Bands of passivity violations have traditionally been identified via the eigenvalues of a Hamiltonian matrix \mathbf{M} that is associated with the state-space model [15], [18], [23]. Since the pole-residue models considered in this work are symmetrical by construction, the same information can be obtained via the eigenvalues of the half-size passivity matrix \mathbf{P} (4), [29]

$$\mathbf{P} = (\mathbf{A} - \mathbf{B}(\mathbf{D} - \mathbf{I})^{-1}\mathbf{C})(\mathbf{A} - \mathbf{B}(\mathbf{D} + \mathbf{I})^{-1}\mathbf{C}). \quad (4)$$

The square-root of any negative-real eigenvalue of \mathbf{P} defines a crossover frequency $j\omega$ where a singular value changes from being smaller than to larger than unity, or vice versa [29]. Note that usage of \mathbf{P} instead of \mathbf{M} gives a reduction in computation time for the eigenvalues by a factor of about eight, since the complexity of eigenvalue computation is cubic.

Frequency bands of passivity violations are next identified by checking the singular values of \mathbf{S} at the midpoint between the crossover frequencies, similarly as shown in [22, Sec. III] for Y -parameter models.

B. Identifying Maxima for Passivity Violations

Within each violating band, the local maxima are identified by a frequency sweep. The singular values as returned by a general SVD algorithm are sorted according to their size. This implies that spurious maxima appear whenever two singular values cross. This problem is overcome by rearranging the sequence of the singular values (and columns in \mathbf{U} and \mathbf{V}) when moving from one frequency point to the next. This is achieved by assessing the change in the direction of the columns of \mathbf{U} , using the switching-back procedure introduced for eigenvalue decompositions in [32].

IV. PERTURBATION

The singular values of \mathbf{S} can be obtained as the eigenvalues of the augmented matrix \mathbf{H} (5) which leads to the eigenvalue problem (6), [33]. The superscript H denotes transpose and conjugate

$$\mathbf{H} = \begin{bmatrix} 0 & \mathbf{S}^H \\ \mathbf{S} & 0 \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \mathbf{V} & \mathbf{V} \\ \mathbf{U} & -\mathbf{U} \end{bmatrix}^{-1} \begin{bmatrix} 0 & \mathbf{S}^H \\ \mathbf{S} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V} & \mathbf{V} \\ \mathbf{U} & -\mathbf{U} \end{bmatrix} = \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix}. \quad (6)$$

Carrying out the matrix inverse gives (7), see Appendix. Retaining the partition associated with the positive Σ leads to (8), which in compact form is written as (9)

$$\frac{1}{2} \begin{bmatrix} \mathbf{V}^{-1} & \mathbf{U}^{-1} \\ \mathbf{V}^{-1} & -\mathbf{U}^{-1} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{S}^H \\ \mathbf{S} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V} & \mathbf{V} \\ \mathbf{U} & -\mathbf{U} \end{bmatrix} = \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix} \quad (7)$$

$$\frac{1}{2} \begin{bmatrix} \mathbf{V}^{-1} & \mathbf{U}^{-1} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{S}^H \\ \mathbf{S} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{U} \end{bmatrix} = \Sigma \quad (8)$$

$$\mathbf{QHT} = \Sigma. \quad (9)$$

The relation between a perturbed singular value and the elements of \mathbf{H} can now be established using the perturbation theory of eigenvalues. First order perturbation of the (truncated) eigenvalue problem (9) leads to the linear relation (10), where \mathbf{q}_i^T and \mathbf{t}_i denote the i th row of \mathbf{Q} and the i th column of \mathbf{T} , respectively

$$\Delta\sigma_i(\mathbf{S}) = \Delta\lambda_i(\mathbf{H}) = \frac{\mathbf{q}_i^T \Delta\mathbf{H} \mathbf{t}_i}{\mathbf{q}_i^T \mathbf{t}_i}, \quad i = 1 \dots n. \quad (10)$$

When the eigenvectors have been scaled to unit length, the denominator of (10) is unity. Introducing the partitioning (11) and carrying out the multiplication gives the sum (12)

$$\Delta\sigma_i = [\mathbf{q}_{i,a}^T \quad \mathbf{q}_{i,b}^T] \begin{bmatrix} 0 & \Delta\mathbf{S}^H \\ \Delta\mathbf{S} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t}_{i,a} \\ \mathbf{t}_{i,b} \end{bmatrix}, \quad i = 1 \dots n \quad (11)$$

$$\Delta\sigma_i = \mathbf{q}_{i,b}^T \Delta\mathbf{S} \mathbf{t}_{i,a} + \mathbf{q}_{i,a}^T \Delta\mathbf{S}^H \mathbf{t}_{i,b}, \quad i = 1 \dots n. \quad (12)$$

It is noted that the two terms in (12) are the conjugate of each other. This gives the simpler form (13)

$$\Delta\sigma_i = 2\mathbf{q}_{i,b}^T \Delta\mathbf{S} \mathbf{t}_{i,a}, \quad i = 1 \dots n. \quad (13)$$

V. PASSIVITY ENFORCEMENT

A. Basic Formulation

The objective is to calculate a minimal perturbation $\Delta\mathbf{S}$ (14a) to the model, such that the passivity constraint (14b) is satisfied

$$\Delta\mathbf{S} \cong 0 \quad (14a)$$

$$\sigma_i + \Delta\sigma_i < 1, \quad i = 1 \dots n \quad (14b)$$

where

$$\Delta\mathbf{S} = \sum_{m=1}^N \frac{\Delta\mathbf{R}_m}{s - a_m} + \Delta\mathbf{D}. \quad (15)$$

Equation (14) is formulated as a constrained least squares problem (16), similarly as in [14], where $\Delta\mathbf{x}$ holds the free variables (elements of $\{\Delta\mathbf{R}_m\}$ and $\Delta\mathbf{D}$). Equation (16) is solved using quadratic programming

$$\min_{\Delta\mathbf{x}} \frac{1}{2} (\Delta\mathbf{x}^T \mathbf{A}_{\text{sys}}^T \mathbf{A}_{\text{sys}} \Delta\mathbf{x}) \quad (16a)$$

$$\mathbf{B}_{\text{sys}} \Delta\mathbf{x} < \mathbf{c}. \quad (16b)$$

B. Diagonalization of Residue Matrices

Each residue matrix is diagonalized (17). Since the residue matrices are real and symmetric, the eigenvector matrices \mathbf{P}_m are real and the matrix inverse equals the matrix transpose

$$\mathbf{R}_m = \mathbf{P}_m \Lambda_{\mathbf{R}_m} \mathbf{P}_m^T. \quad (17)$$

First order perturbation of a residue matrix gives

$$\frac{\mathbf{R}_m + \Delta\mathbf{R}_m}{s - a_m} \approx \frac{\mathbf{P}_m (\Lambda_{\mathbf{R}_m} + \Delta\Lambda_{\mathbf{R}_m}) \mathbf{P}_m^T}{s - a_m}. \quad (18)$$

The perturbed residue matrix eigenvalues $\Delta\Lambda_{\mathbf{R}_m}$ are chosen as free variables in (16) via (19a), rather than the (perturbed) residue matrix elements. In the case of a complex conjugate pair, the real and imaginary parts are diagonalized separately. Similarly, the eigenvalues of a nonzero \mathbf{D} are used as free variables (19b). Note that the eigenvector matrices \mathbf{P}_m and \mathbf{P}_D are obtained directly from \mathbf{R}_m and \mathbf{D} , which are known quantities

$$\Delta\mathbf{R}_m = \mathbf{P}_m \Delta\Lambda_{\mathbf{R}_m} \mathbf{P}_m^T \quad (19a)$$

$$\Delta\mathbf{D} = \mathbf{P}_D \Delta\Lambda_D \mathbf{P}_D^T. \quad (19b)$$

During passivity enforcement, all residue matrix elements become perturbed via a reduced set of variables. The number of free variables is nN compared to $n(n+1)N/2$ when individually perturbing all matrix elements and utilizing symmetry. As a result, the computation time needed for solving (16) is greatly reduced since the computation time of the basic operations in quadratic programming is cubic with problem size. After solving (16), the changes $\{\Delta\mathbf{R}_m\}$ and $\Delta\mathbf{D}$ are recovered via (19a) and (19b).

C. Building the Least Squares Equation

First, consider a pole-residue term associated with a real pole. Expanding the factorization (19a) into a sum of outer products gives (20), where \mathbf{p}_j is the j th column of \mathbf{P} in (19a)

$$\begin{aligned} \frac{\Delta\mathbf{R}}{s - a} &= \frac{1}{s - a} \sum_{j=1}^n \mathbf{p}_j \mathbf{p}_j^T \Delta\lambda_j \\ &= \frac{1}{s - a} \sum_{j=1}^n \Gamma_j \Delta\lambda_j \cong 0. \end{aligned} \quad (20)$$

By placing the elements of $\Delta\mathbf{R}$ in a single vector, (20) can be cast in the form of a matrix-vector equation with $\Delta\lambda_j$ in the vector of unknowns. This is shown in (21) for an example with $n = 2$ ports. (The superscript denotes “ j ” in (20) while the subscript denotes the matrix entry). It is observed that enforcement of symmetry leads to the deletion of one row (elements (2,1)) and to the scaling of one row (elements (1, 2)) with a factor of two

$$\frac{1}{s - a} \begin{bmatrix} \Gamma_{1,1}^1 & \Gamma_{1,1}^2 \\ 2\Gamma_{1,2}^1 & 2\Gamma_{1,2}^2 \\ \Gamma_{2,2}^1 & \Gamma_{2,2}^2 \end{bmatrix} \begin{bmatrix} \Delta\lambda_1 \\ \Delta\lambda_2 \end{bmatrix} \cong 0. \quad (21)$$

Next, consider one complex conjugate pair. In order to enforce the conjugacy requirement for $\mathbf{S}(s)$ in the solution, the pair is separated into its real and imaginary parts (22)

$$\begin{aligned} \frac{\Delta\mathbf{R}' + j\Delta\mathbf{R}''}{s - (a' + ja'')} + \frac{\Delta\mathbf{R}' - j\Delta\mathbf{R}''}{s - (a' - ja'')} \\ = f(s)\Delta\mathbf{R}' + g(s)\Delta\mathbf{R}'' \end{aligned} \quad (22a)$$

$$f(s) = \frac{1}{s - (a' + ja'')} + \frac{1}{s - (a' - ja'')} \quad (22b)$$

$$g(s) = \frac{j}{s - (a' + ja'')} - \frac{j}{s - (a' - ja'')}. \quad (22c)$$

Combining (22) with the eigenvalue decomposition (20) leads to (23), which is written as a matrix equation, similarly as in (21)

$$f(s) \sum_{j=1}^n \mathbf{\Gamma}'_j \Delta \lambda'_j + g(s) \sum_{j=1}^n \mathbf{\Gamma}''_j \Delta \lambda''_j \cong 0. \quad (23)$$

Equation (23) is recasted into the form of a system matrix \mathbf{A}_{sys} (24), where the free variables are enforced to become real quantities by separating each equation into its real and imaginary parts

$$\mathbf{A}_{\text{sys}} \Delta \tilde{\mathbf{x}} = \begin{bmatrix} \Re \mathbf{A} \\ \Im \mathbf{A} \end{bmatrix} \Delta \tilde{\mathbf{x}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (24)$$

\mathbf{A}_{sys} has $Mn(n+1)$ rows and Nn columns, where M is the number of frequency samples, n is the number of ports, and N is the order. In the case of applications with many ports (e.g., packages), the memory requirements for \mathbf{A}_{sys} can become excessive. This problem is avoided by building the normal equations $\mathbf{A}_{\text{sys}}^T \mathbf{A}_{\text{sys}}$ of (16a) via a sum of outer products (25), thereby avoiding the need for forming \mathbf{A}_{sys}

$$\mathbf{A}_{\text{sys}}^T \mathbf{A}_{\text{sys}} = \sum_{\text{row}} \mathbf{A}_{\text{sys,row}}^T \mathbf{A}_{\text{sys,row}}. \quad (25)$$

D. Building the Constraint Equation

In the case of a real pole-residue term, (20) replaces $\Delta \mathbf{S}$ in (13). This gives the relation (26) between a perturbation of a free variable $\Delta \lambda_j$ and the singular value, $\Delta \sigma$. (The subscript i in (13) has been discarded for clarity). The computation of the coefficients in (26) is fast since it involves inner products

$$\Delta \sigma = \frac{2 \sum_{j=1}^n (\mathbf{q}_b^T \mathbf{p}_j) (\mathbf{p}_j^T \mathbf{t}_a) \Delta \lambda_j}{s - a}. \quad (26)$$

In the case of a complex conjugate pair, the contribution from the j th eigenvalue is shown in (27). The prime and double prime denote the contribution from $\Delta \mathbf{R}'$ and $\Delta \mathbf{R}''$, respectively

$$\Delta \sigma = [k_1(s) \quad k_2(s)] \begin{bmatrix} \Delta \lambda'_j \\ \Delta \lambda''_j \end{bmatrix} \quad (27a)$$

$$k_1(s) = 2f(s) (\mathbf{q}_b^T \mathbf{p}'_j) (\mathbf{p}'_j^T \mathbf{t}_a) \quad (27b)$$

$$k_2(s) = 2g(s) (\mathbf{q}_b^T \mathbf{p}''_j) (\mathbf{p}''_j^T \mathbf{t}_a). \quad (27c)$$

The presence of a proportional term \mathbf{D} is handled as in (21) and (26), but with the coefficient $(s - a)$ replaced with unity.

Finally, the resulting constraint matrix \mathbf{B}_{sys} in (16b) is replaced with its real part.

E. Column Scaling

The conditioning of (16) is improved by scaling the columns of \mathbf{A}_{sys} . Normally, these columns would be scaled to unit Euclidian length. However, since the columns of \mathbf{A}_{sys} are not explicitly formed (Section IV-C), the scaling is unknown. As appropriate scaling we therefore choose the inverse norm of the

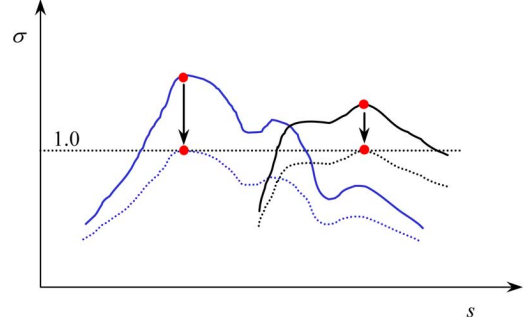


Fig. 1. Passivity enforcement at singular value maxima.

basis functions. This implies the scaling $\|(s - a)\|_2$ for a real pole, and $\|1/f(s)\|_2$ and $\|1/g(s)\|_2$ in the case of a complex pair, with f and g defined in (22). After solving (16), the elements of the solution vector are recovered by multiplication with the same scaling factors.

F. Sample Selection

The frequency samples for the least squares part (16a) can in principle be arbitrarily chosen. It does, however, make more sense to use the same samples as was used in the identification of the original pole-residue model (1). This choice has the additional advantage that it leads to adequate numerical conditioning of (16a). In order to further improve the conditioning, auxiliary frequency samples are added at frequencies which correspond to the location of out-of-band poles. These samples are given a low weighting in the least squares problem so that they do in practice not affect the in-band result.

Frequency samples for the constraint part (16b) are chosen as maxima for violating singular values, see Fig. 1. This reduces the number of rows in \mathbf{B}_{sys} to a minimum, thereby increasing the computational efficiency. The singular values are enforced to be smaller than unity by a small value, in order to reduce the required number of iterations.

G. Error Control Strategies

Each row in \mathbf{A}_{sys} in (16a) corresponds to one element of \mathbf{S} at one frequency sample. The quality of the perturbed model can therefore be manipulated by row-weighting. For instance, weighting with the inverse element magnitude is a straightforward approach for achieving relative error control.

H. Iterations

The passivity enforcement step must be applied repeatedly due to the nonlinearity of (16b), and because the passivity enforcement step may cause new violations to arise. In practice, divergence may occur. The divergence problem is overcome by the robust iteration procedure introduced in [17] and [21] which employs an inner loop that adds more constraints at frequencies where new passivity violations occur. This feature was not needed for the examples in this paper.

\mathbf{A}_{sys} is built during the first iteration and reused in the subsequent iterations, thereby saving computation time. This implies that the first iteration is computationally more expensive than the rest.

- Convert pole-residue model ($\{\mathbf{R}_m\}, \mathbf{D}$) into state-space model ($\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$) (1).
- Form passivity matrix, \mathbf{P} (4).
- Identify crossover frequencies for singular values of \mathbf{S} as the square-root of negative eigenvalues of \mathbf{P} .
- From list of crossover-frequencies, identify bands of passivity violations.
- Identify frequencies of maximum violation (Fig.1) by sweeping singular values of \mathbf{S} .

Fig. 2. Passivity assessment.

- Calculate \mathbf{S} at frequencies of maximum violation.
 - Calculate singular value decomposition, $[\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}] = \text{svd}(\mathbf{S})$.
 - Form $\mathbf{q}_{i,b}^T$ and $\mathbf{t}_{i,a}^T$ of the first order perturbation (13).
 - Diagonalize matrices $\{\mathbf{R}_m\}$ and \mathbf{D} (19).
 - Take the eigenvalues as free variables and calculate the reduced-size perturbation (26).
 - Build the (reduced) constraint equation (16b).
- If first iteration step:
- Calculate the reduced-size cost function (23) at frequencies defining the fitting band.
 - Build the (reduced) cost equation (16a).
- Solve (16) using Quadratic Programming.
 - Update pole-residue model.

Fig. 3. Passivity enforcement.

VI. ALGORITHM OVERVIEW

The main steps in the procedure for passivity assessment (Section III) and passivity enforcement (Sections IV and V) are summarized in Figs. 2 and 3, respectively.

VII. TEST CASES

The complete passivity enforcement procedure is applied to four different test cases: a microwave filter, a high-speed interconnect, and two package applications. In all cases, the usage of an inner loop (V-H) is disabled. The listed CPU-times *include all overhead costs*. The CPU-times for critical computations are listed separately in Section VIII. All computations are with Matlab running on a desktop with a 1.3 GHz Pentium processor and 2 Gb RAM. The quadratic programming problem (16) is solved using routine “quadprog” in the Matlab Optimization Toolbox. The singular value decomposition of \mathbf{S} is calculated using routine “svd” in Matlab, which is based on LAPACK routine ZGESVD [34] (complex and double precision). This routine makes use of bidiagonalization followed by singular value decomposition by a QR-based procedure described in [35].

A. Case I: Microwave Filter

The first example is a tenth-order rational model of a microwave hairpin filter [20]. As shown in Fig. 4, the model has singular values exceeding unity, thus being nonpassive.

Passivity enforcement was applied with relative error control by usage of inverse least squares weighting in (16a). It is seen in Fig. 4 that all singular values become smaller than unity, thus implying a passive model.

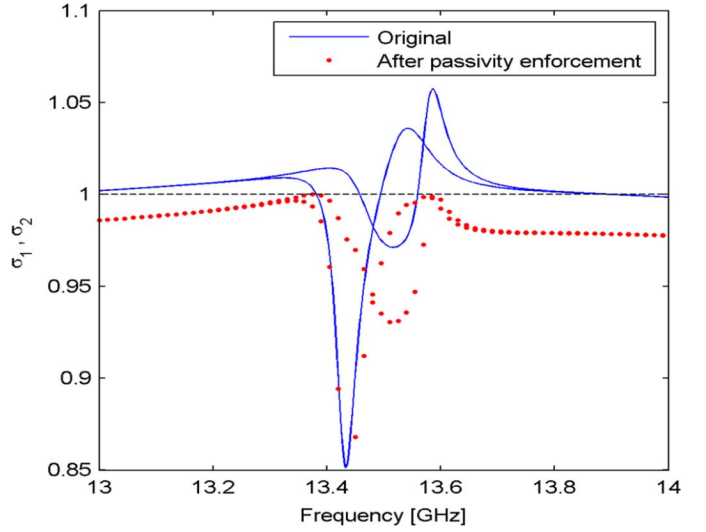
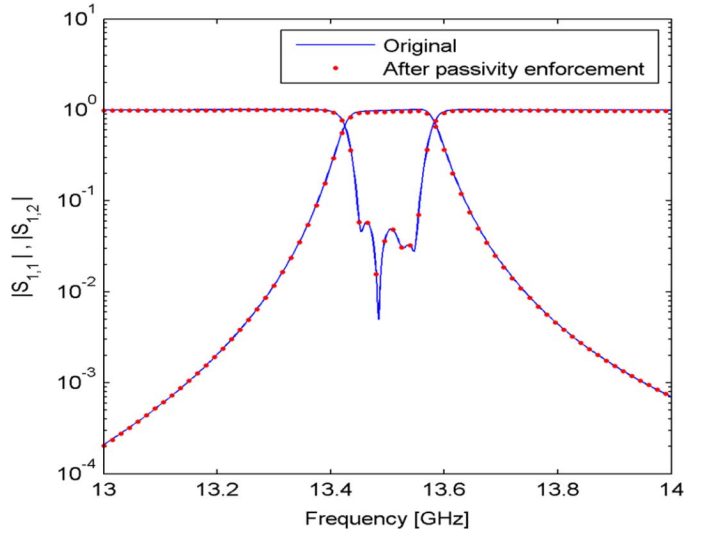
Fig. 4. Singular values of \mathbf{S} .Fig. 5. Elements of \mathbf{S} .

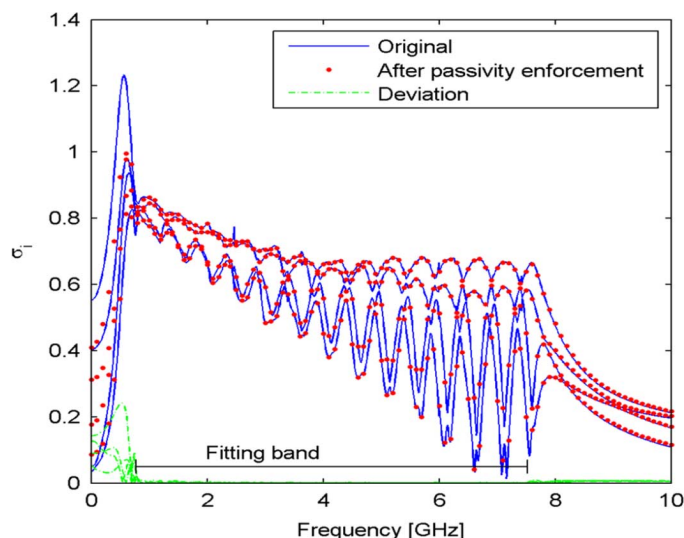
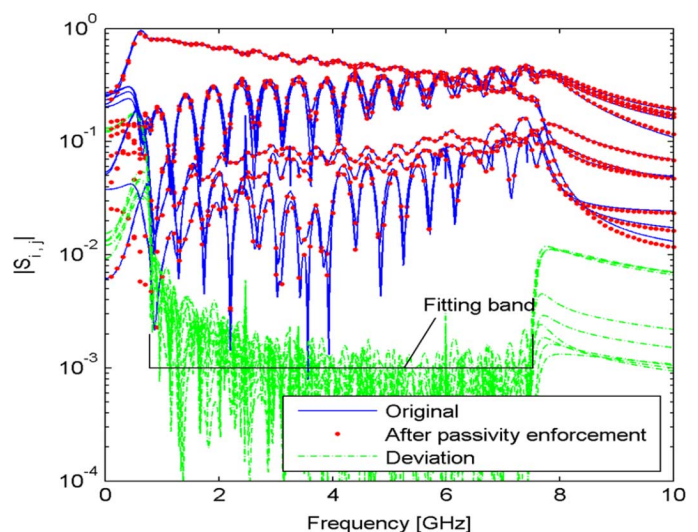
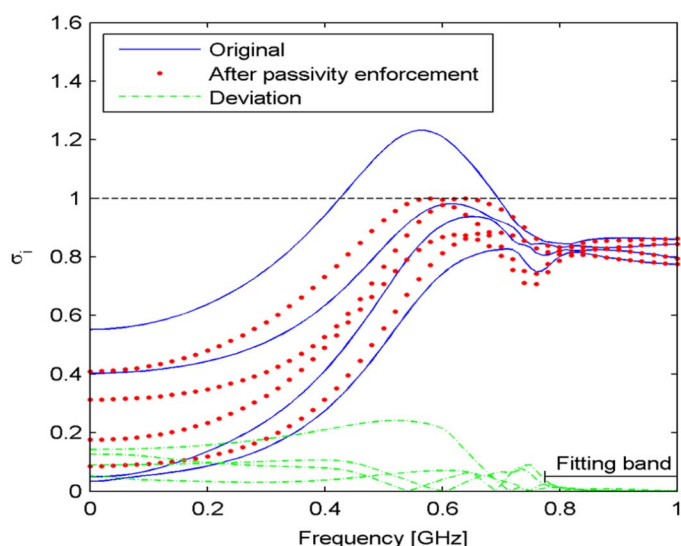
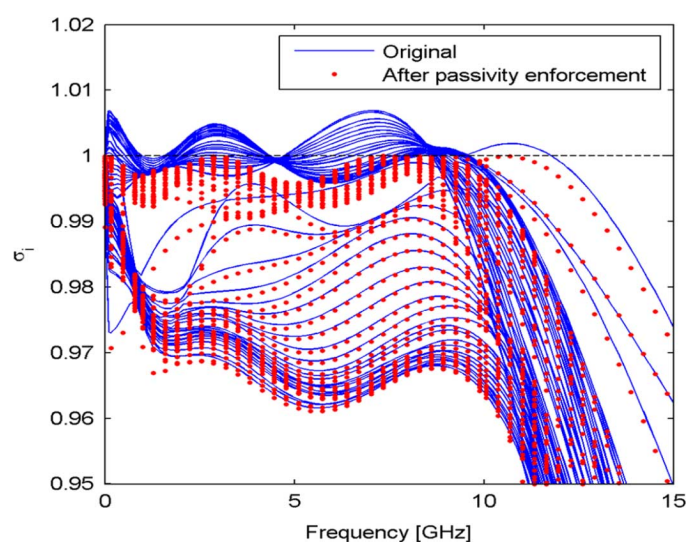
Fig. 5 shows the elements of \mathbf{S} . It is seen that the passivity enforcement does not impair the accuracy of the model. It is noted that with unitary weights (not shown), the small elements in Fig. 5 were inaccurate where they are smaller than 0.001 in magnitude.

The passivity enforcement required three iterations, using a total of 3.0 s (including passivity assessment).

B. Case II: Interconnect System

This example considers a chip-to-chip interconnect system whose scattering parameters have been measured in the frequency domain [36]. This four-port system is subjected to rational fitting by vector fitting in the frequency range model 0.775–7.52 GHz using 100 pole-residue terms and 271 frequency samples.

The resulting model has a large passivity violation at low frequencies (out-of-band). Figs. 6 and 7 show the effect of the passivity enforcement on the singular values of \mathbf{S} , using unitary weighting. It is seen that passivity is enforced with only a

Fig. 6. Singular values of S .Fig. 8. Elements of S .Fig. 7. Singular values of S . Low frequencies.Fig. 9. Singular values of S .

small perturbation of the in-band behavior. This is clearly seen in Fig. 8, which shows the effect of the passivity enforcement on the elements of S . (In all plots, “Deviation” denotes the magnitude of the complex-valued deviation, thus taking into account the phase difference.)

The passivity enforcement required four iterations, using a total of 22 s (10 s for passivity assessment and 12 s for passivity enforcement).

C. Case III: Low-Order Package Model

This example considers a nonpassive rational model of a ball grid array package [20]. The model has 48 ports and is represented by six pole-residue terms and a constant term (D).

The model has substantial passivity violations, see Fig. 9. The passivity enforcement (using weighting with the inverse element magnitude) required six iterations, using a total of 140 s (62 s for passivity assessment and 78 s for passivity enforcement). The result in Fig. 9 shows that all singular values have been enforced to be smaller than unity.

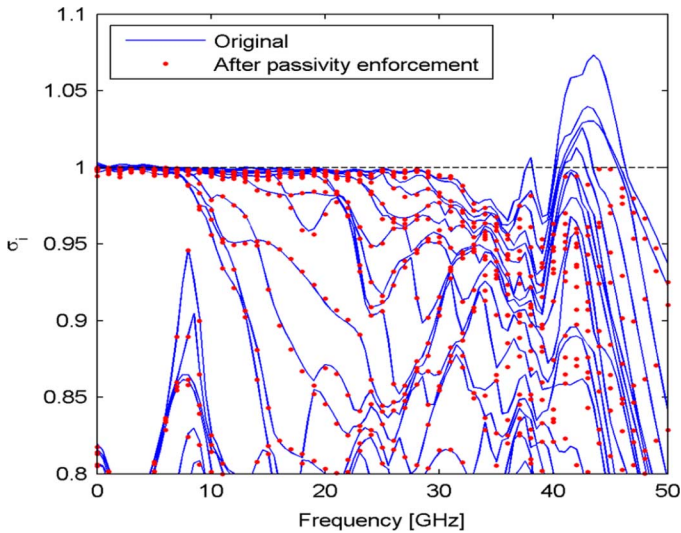
D. Case IV: High-Order Package Model

This 28-port example considers the modeling of a surface mount package [18]. The extracted model has 40 pole-residue terms and a constant term (D). Thus, A is quite large (1120×1120).

The model has passivity violations at both low and high frequencies, see Fig. 10. The passivity enforcement (using weighting with the inverse element magnitude) required three iterations, using a total of 466 s (148 s for passivity assessment and 318 s for passivity enforcement). It is observed that all singular values have been enforced to be smaller than unity.

VIII. COMPUTATIONAL EFFICIENCY

In the following we compare the computational efficiency of the proposed implementation with that of other implementations. As before, all calculations are with a 1.3-GHz Pentium Processor. The focus is on the critical computational steps: Calculating crossover frequencies (passivity assessment) and

Fig. 10. Singular values of S .TABLE I
CASE DESCRIPTION

Model	Poles M	Ports n	Description
Case I	20	2	MW filter
Case II	400	4	Interconnect
Case III	288	48	BGA package
Case IV	1120	28	Surface mount package

TABLE II
CPU TIME FOR EIGENVALUE COMPUTATION

Model	$\text{eig}(M)$	$\text{eig}(P)$	Speedup
Case II	6.9 sec	0.84 sec	8.2
Case III	2.2 sec	0.24 sec	9.2
Case IV	111 sec	14.7 sec	7.6

solving the quadratic programming problem (16) (passivity enforcement).

Table I summarizes the characteristics of the test cases. The quantity M denotes the number of poles of the associated state-space model, $M = nN$.

Table II compares the CPU time for eigenvalue computation when the crossover frequencies are obtained from either the traditional Hamiltonian matrix M or the half-size passivity matrix P as adopted in this work. It is observed that usage of P leads to about eight times faster computation of eigenvalues.

Table III lists three alternative methods for passivity enforcement.

- A) The perturbed residue matrix elements are taken as free variables while utilizing symmetry [14]. The (sparse) quadratic programming (QP) problem (16) is solved

TABLE III
PASSIVITY ENFORCEMENT METHODS

Method	Description	QP solver
A	Residue perturbation.	quadprog
B	Residue perturbation.	CPLEX
C	Eigenvalue perturbation. (method in this paper)	quadprog

TABLE IV
NUMBER OF ITERATIONS

Model	A	B	C
Case II	3	3	4
Case III	—	3	6
Case IV	—	4	3

TABLE V
COMPUTATION TIME FOR SOLVING QP PROBLEMS

Model	A	B	C
Case II	20.1 sec	1.1 sec	4.4 sec
Case III	—	1.3 min	0.4 min
Case IV	—	1.5 min	2.7 min

using routine “quadprog” in the Matlab Optimization Toolbox. quadprog does not utilize the sparsity of (16a).

- B) Same as A , but with the QP-problem solved using CPLEX [37]. CPLEX takes advantage of the sparsity of (16a), unlike quadprog.
- C) Method used in this paper. The perturbed residue matrix eigenvalues are taken as free variables. The QP problem (16) is solved using the Matlab routine “quadprog.”

Table IV shows that all methods A – C successfully enforce passivity in 3–6 iterations. Method A is not applicable to problem cases III and IV due to excessive memory requirements.

Table V lists the total CPU time (all iterations in Table IV) needed for solving the QP-problems. In Case II (interconnect), method A requires 20.1 s while methods B and C solve the problems in only 1.1 s and 4.4 s. In Case III (BGA package), method B requires 1.3 min with C requiring only 0.4 min. In Case IV, method B is the fastest. It can be concluded that method C represents a good alternative to method B . It has the additional advantage of not needing a sparse solver (e.g., CPLEX) for the QP-problem. On the other hand, the resulting perturbation of the model is often smaller with method B , due to the higher number of free variables. Fig. 11 shows the elements of the scattering matrix S for Case II when passivity has been enforced using method B . Comparison with Fig. 8 (method C) shows that the change to the model is smaller.

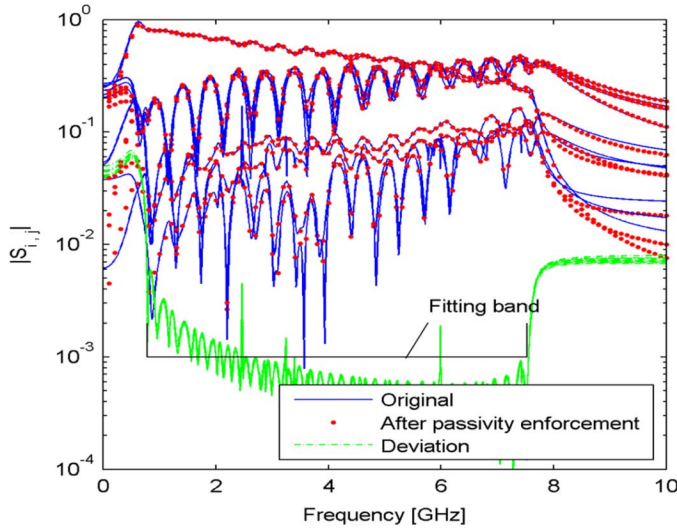


Fig. 11. Elements of S . Passivity enforcement by residue perturbation. (Method B in Table III).

Note that the passivity assessment has to be carried out in every iteration. For instance, in Case III with method C , a total of seven passivity checks is done, requiring a total of $7 \times 0.24 = 1.7$ s.

IX. DISCUSSION

Using perturbed residue matrix eigenvalues as free variables leads to a significant speed improvement over the approach with direct perturbation of residue matrix elements [14] because the number of free elements is reduced from $Nn(n+1)/2$ to Nn . Since the complexity of the basic steps in quadratic programming (used for solving 16) is cubic, it follows that the complexity is reduced from $O(N^3n^6)$ to $O(N^3n^3)$. Thus, the speed improvement is particularly significant for cases with many ports, n , typically found in package applications. Direct perturbation of residue matrix elements also leads to an efficient approach when utilizing the sparsity of the problem formulation (Section VIII, method B). That approach, however, requires the usage of a specialized sparse solver for the quadratic programming problem (16), e.g., CPLEX. Such solvers are often expensive.

X. CONCLUSION

A passivity enforcement method has been introduced for use with symmetrical S -parameter rational macromodels on pole-residue form.

- 1) Passivity is enforced by perturbing the residues such that the singular values become smaller than unity at frequency points where the singular values are maximum and greater than unity.
- 2) The approach is efficient in terms of CPU time and memory requirements since the perturbed eigenvalues of the residue matrices are used as free variables, rather than the (perturbed) residue matrix elements. This reduces the size of the quadratic programming problem to be solved, in particular for cases with many ports.

- 3) Error control strategies are easily implemented via least squares weighting, for instance relative error control.
- 4) The passivity assessments is accurately and efficiently carried out by checking the eigenvalues of a test matrix which is half the size of the traditional Hamiltonian matrix.
- 5) The approach was successfully applied to four test examples: a microwave filter, a high-speed interconnect structure, and two package applications.
- 6) The computational efficiency is comparable to what can be achieved with direct perturbation of residue matrix elements combined with a sparse solver (CPLEX).

APPENDIX

INVERSION OF PARTITIONED MATRIX

For a matrix partitioned into four blocks (28), the matrix inverse can be calculated by (29), [38]

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (28)$$

$$A^{-1} = \begin{bmatrix} C_1^{-1} & -A_{11}^{-1}A_{12}C_2^{-1} \\ -C_2^{-1}A_{21}A_{11}^{-1} & C_2^{-1} \end{bmatrix} \quad (29)$$

where

$$C_1 = A_{11} - A_{12}A_{22}^{-1}A_{21} \quad (30a)$$

$$C_2 = A_{22} - A_{21}A_{11}^{-1}A_{12}. \quad (30b)$$

In (6), we are interested in the matrix inverse of

$$A = \begin{bmatrix} V & V \\ U & -U \end{bmatrix} \quad (31)$$

This gives for (30a), (30b)

$$C_1 = 2V \quad (32a)$$

$$C_2 = -2U \quad (32b)$$

and for (29)

$$A^{-1} = \frac{1}{2} \begin{bmatrix} V^{-1} & U^{-1} \\ V^{-1} & -U^{-1} \end{bmatrix}. \quad (33)$$

ACKNOWLEDGMENT

The author would like to thank the following individuals for providing test examples: A. Lamecki (Gdansk University of Technology, Gdansk, Poland) for microwave filter example in Section VII-A and package example in Section VII-C; Dr. W. T. Beyene (Rambus Inc., Los Altos, CA) for interconnect example in Section VII-B; and Dr. S. Grivet-Talocia (Politecnico di Torino, Italy) for package example in Section VII-D.

REFERENCES

- [1] R. Achar and M. Nakhla, "Simulation of high-speed interconnects," *Proc. IEEE*, vol. 89, no. 5, pp. 693–728, May 2001.

- [2] G. Antonini, "SPICE equivalent circuits of frequency-domain responses," *IEEE Trans. Electromagn. Compat.*, vol. 45, no. 3, pp. 502–512, Aug. 2003.
- [3] A. Semlyen and A. Dabuleanu, "Fast and accurate switching transient calculations on transmission lines with ground return using recursive convolutions," *IEEE Trans. Power App. Syst.*, vol. 94, no. 2, pp. 561–575, Mar./Apr. 1975.
- [4] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1052–1061, Jul. 1999.
- [5] S. Grivet-Talocia, "Package macromodeling via time-domain vector fitting," *IEEE Microwave Wireless Compon. Lett.*, vol. 13, no. 11, pp. 472–474, Nov. 2003.
- [6] B. Gustavsen, "Improving the pole relocating properties of vector fitting," *IEEE Trans. Power Del.*, vol. 21, no. 3, pp. 1587–1592, Jul. 2006.
- [7] D. Deschrijver, B. Haegeman, and T. Dhaene, "Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses," *IEEE Trans. Adv. Packag.*, vol. 30, no. 2, pp. 216–225, May 2007.
- [8] Y. S. Mekonnen and J. E. Schutt-Ainé, "Broadband macromodeling of sampled frequency data using z-domain vector-fitting method," in *Proc. 11th IEEE Workshop Signal Propagat. Interconnects*, Ruta de Camogli, Italy, May 13–16, 2007, pp. 45–48.
- [9] N. Wong and C.-U. Lei, "IIR approximation of FIR filters via discrete-time vector fitting," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1296–1032, Mar. 2008.
- [10] D. Deschrijver, T. Dhaene, and D. De Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *IEEE Microwave Wireless Compon. Lett.*, vol. 18, no. 6, pp. 383–385, Jun. 2008.
- [11] B. Gustavsen and C. Heitz, "Modal vector fitting: A tool for generating rational models of high accuracy with arbitrary terminal conditions," *IEEE Trans. Adv. Packag.*, vol. 31, no. 4, pp. 664–672, Nov. 2008.
- [12] C. P. Coelho, J. Phillips, and L. M. Silveira, "A convex programming approach for generating guaranteed passive approximations to tabulated frequency-data," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 23, no. 2, pp. 293–301, Feb. 2004.
- [13] S. Grivet-Talocia and A. Ubolli, "A comparative study of passivity enforcement schemes for linear lumped macromodels," *IEEE Trans. Adv. Packag.*, vol. 31, no. 4, pp. 673–683, Nov. 2008.
- [14] B. Gustavsen and A. Semlyen, "Enforcing passivity for admittance matrices approximated by rational functions," *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 97–104, Feb. 2001.
- [15] D. Saraswat, R. Achar, and M. S. Nakhla, "Global passivity enforcement algorithm for macromodels of interconnect subnetworks characterized by tabulated data," *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 13, no. 7, pp. 819–832, Jul. 2005.
- [16] D. Saraswat, R. Achar, and M. Nakhla, "On passivity enforcement for macromodels of S-parameter based tabulated subnetworks," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 23–26, 2005, pp. 3777–3780.
- [17] B. Gustavsen, "Computer code for passivity enforcement of rational macromodels by residue perturbation," *IEEE Trans. Adv. Packag.*, vol. 30, no. 2, pp. 209–215, May 2007.
- [18] S. Grivet-Talocia, "Passivity enforcement via perturbation of Hamiltonian matrices," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 9, pp. 1755–1769, Sep. 2004.
- [19] S. Grivet-Talocia and A. Ubolli, "Passivity enforcement with relative error control," *IEEE Trans. Microwave Theory Tech.*, vol. 55, no. 11, pp. 2374–2383, Nov. 2007.
- [20] A. Lamecki and M. Mrozowski, "Equivalent SPICE circuits with guaranteed passivity from nonpassive models," *IEEE Trans. Microwave Theory Tech.*, vol. 55, no. 3, pp. 526–532, Mar. 2007.
- [21] B. Gustavsen, "Fast passivity enforcement for pole-residue models by perturbation of residue matrix eigenvalues," *IEEE Trans. Power Del.*, vol. 23, no. 4, pp. 2278–2285, Oct. 2008.
- [22] B. Gustavsen, "Passivity enforcement of rational models via modal perturbation," *IEEE Trans. Power Del.*, vol. 23, no. 2, pp. 768–775, Apr. 2008.
- [23] S. Boyd, V. Balakrishnan, and P. Kabamba, "A bisection method for computing the H_∞ norm of a transfer matrix and related problems," *Math. Control. Signals, Syst.*, vol. 2, no. 3, pp. 207–219, 1989.
- [24] S. Grivet-Talocia and A. Ubolli, "On the generation of large passive macromodels for complex interconnect structures," *IEEE Trans. Adv. Packag.*, vol. 29, no. 1, pp. 39–54, Feb. 2006.
- [25] D. Saraswat, R. Achar, and M. S. Nakhla, "Fast passivity verification and enforcement via reciprocal systems for interconnects with large order macromodels," *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 15, no. 1, pp. 48–59, Jan. 2007.
- [26] S. Grivet-Talocia, "An adaptive sampling technique for passivity characterization and enforcement of large interconnect macromodels," *IEEE Trans. Adv. Packag.*, vol. 30, no. 2, pp. 226–237, May 2007.
- [27] A. Semlyen and B. Gustavsen, "A half size singularity test matrix for fast and reliable passivity assessment of rational models," *IEEE Trans. Power Del.*, vol. 24, no. 1, pp. 345–351, Jan. 2009.
- [28] B. Gustavsen and A. Semlyen, "On passivity tests for unsymmetrical models," *IEEE PES Lett.*, to be published.
- [29] B. Gustavsen and A. Semlyen, "Fast passivity assessment for S-parameter rational models via a half-size test matrix," *IEEE Trans. Microwave Theory Tech.*, vol. 56, no. 12, pp. 2701–2708, Dec. 2008.
- [30] M. R. Wohlers, *Lumped and Distributed Passive Networks*. New York: Academic, 1969.
- [31] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. G. Canavero, and R. Achar, "Stability, causality, and passivity in electrical interconnect models," *IEEE Trans. Adv. Packag.*, vol. 30, no. 4, pp. 795–808, Nov. 2007.
- [32] L. M. Wedepohl, H. V. Nguyen, and G. D. Irwin, "Frequency-dependent transformation matrices for untransposed transmission lines using Newton-Raphson method," *IEEE Trans. Power Del.*, vol. 11, no. 3, pp. 1538–1546, Aug. 1996.
- [33] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. Philadelphia, PA: SIAM, 1997, p. 235.
- [34] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK User's Guide*, 3rd ed. Philadelphia, PA: SIAM, 1999.
- [35] J. W. Demmel and W. Kahan, "Accurate singular values of bidiagonal matrices," *SIAM J. Sci. Stat. Comput.*, vol. 11, pp. 873–912, 1990.
- [36] W. T. Beyene, J. Feng, N. Cheng, and X. Yuan, "Performance analysis and model-to-hardware correlation of multigigahertz parallel bus with transmit pre-emphasis equalization," *IEEE Trans. Microwave Theory and Techniques*, vol. 53, no. 11, pp. 3568–3577, Nov. 2005.
- [37] CPLEX. Tomlab Optimization Inc., San Diego, CA [Online]. Available: <http://tomopt.com/>
- [38] V. Strassen, "Gaussian elimination is not optimal," *Numer. Math.*, vol. 13, pp. 354–356, 1969.

Bjørn Gustavsen (M'94–SM'03) was born in Harstad, Norway, in 1965. He received the M.Sc. degree and the dr.ing. degree, both from the Norwegian Institute of Technology (NTH), Trondheim, Norway, in 1989 and 1993, respectively.

Since 1994 he has been with SINTEF Energy Research. His interests include simulation of electromagnetic transients and modeling of frequency dependent effects. He spent 1996 as a Visiting Researcher at the University of Toronto, Toronto, ON, Canada, and the summer of 1998 at the Manitoba HVDC Research Centre, Winnipeg, MB, Canada. He was a Marie Curie Fellow at the University of Stuttgart, Stuttgart, Germany from August 2001 to August 2002.