

# Passivity Enforcement by Residue Perturbation via Constrained Non-Negative Least Squares

Bjørn Gustavsen , Fellow, IEEE

**Abstract**—Passivity is a fundamental requirement for rational models to guarantee stable behavior when included in general time domain simulations. This paper introduces a new variant of one passivity enforcement scheme that is based on residue matrix perturbation. The constrained least squares problem associated with the passivity condition is transformed into a least distance problem which is solved as a non-negative least squares problem while utilizing the sparsity pattern that arises with multi-terminal problems. The resulting method is very fast and requires only a small amount of computer memory, thereby being applicable to models with many terminals and high orders. The method is demonstrated to be suitable for frequency-dependent modeling of subnetworks, transformer winding branch impedances, and measured transformer admittance data.

**Index Terms**—Passivity enforcement, residue perturbation, non-negative least squares, FDNE, transformer modeling.

## I. INTRODUCTION

FREQUENCY dependent modeling has over the years become widely applied in electromagnetic transient (EMT) programs. The most common usage is in traveling wave models for representation of the propagation characteristics [1]–[3]. Another emerging application is the representation of subnetworks by a frequency dependent network equivalent (FDNE) to save computation time [4]–[9]. In the latter application, the typical modeling procedure is to generate frequency samples representing the terminal (port) admittance, followed by model extraction via rational function approximation. Finally, the model is subjected to passivity enforcement by perturbation of the model's parameters, thereby ensuring stable time domain simulation results with any terminal condition. While the initial model can be reliably calculated using methods such as vector fitting [10] or Löwner tangential interpolation [11], [12], the passivity enforcement is much more difficult and has hampered the widespread use of FDNEs in electromagnetic transient simulations. The lack of a good passivity enforcement method is also a major

obstacle in some other modeling applications, e.g. transformer black-box modeling from frequency sweep measurements [13], white-box transformer modeling with inclusion of frequency-dependent branch impedance effects [14], and grounding mat frequency-dependent modeling [15].

The passivity enforcement of a model requires the ability to 1) detect and quantify passivity violations, and 2) remove the passivity violations. While the assessment of passivity violations can be achieved by frequency sweeping [16] or by checking eigenvalues of test matrices [17], [18] the passivity enforcement itself is more difficult. A general procedure for passivity enforcement was introduced in [19] where the model's residue matrices are perturbed such that the passivity violations are removed. The mathematical formulation involves a constrained linear least squares problem that can be solved as a quadratic programming (QP) problem. This residue perturbation method (RP) requires iterations because the relation between residue perturbation and the passivity condition is non-linear. The experience with RP is that it can be prohibitively slow in the case of problems with many terminals and/or high model orders.

Several modifications to the RP method have been proposed to improve its performance. The calculation time is reduced when perturbing only a limited number of residue matrices [20], [9] or a limited number of elements in each residue matrix [20], but such approach is heuristic and can involve loss of accuracy in the final model. The lack of a sparse solver for the quadratic problem solver in early Matlab versions inspired the development of the "Fast RP" method (FRP) where the eigenvalues of the residue matrices are perturbed [21]. This reduces the number of free variables, but the resulting method remains slow in the case of models with many terminals. Other improvements have been proposed that reduce large passivity violations via local perturbations [22], but such approach is not general and must often be followed by a final refinement using a general passivity method. Other methods are based on enforcing the passivity by perturbing eigenvalues of a passivity test matrix, [17], [23] but those methods are limited to medium size problems because the calculation of these eigenvalues is prohibitively slow for large models.

In this paper, an alternative way of solving the residue perturbation (RP) problem is introduced that is computationally more efficient than usage of QP. The original RP problem is via QR-decomposition transformed into a least distance problem (LDP) that is solved as a non-negative least squares (NNLS) problem [24]. The computational effort of the QR-decomposition is with multi-terminal problems reduced by utilizing the sparsity

Manuscript received February 26, 2020; revised June 11, 2020, August 10, 2020, and September 10, 2020; accepted September 20, 2020. Date of publication September 24, 2020; date of current version September 23, 2021. This work was supported by the Research Council of Norway and the consortium participants of the SINTEF-led projects "ProTrafo" (project no. 269303/E20) and "FastTrans" (project no. 294508/E20). Paper number TPWRD-00299-2020. (Corresponding author: Bjørn Gustavsen.)

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

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TPWRD.2020.3026385>.

Digital Object Identifier 10.1109/TPWRD.2020.3026385

pattern of the least squares (LS) system matrix. To further enhance the computational speed, the implemented method also permits to perturb subsets of residue matrices, and/or a subset of residue matrix elements. The performance of the resulting method (RP-NNLS) is demonstrated for FDNE modeling, FEM-based white-box transformer branch impedance modeling, and measurement-based transformer black-box modeling.

## II. PROBLEM STATEMENT

The model to be passivated is a pole-residue model (1) that represents a terminal admittance matrix  $\mathbf{Y}$ . The admittance matrix defines the relation between  $n$  terminal voltages  $\mathbf{v}$  and ditto terminal currents  $\mathbf{i}$  (2).  $\mathbf{Y}$  is symmetrical and the poles  $a_i$  are stable. The poles and residue matrices  $\mathbf{R}_i$  are real or complex conjugate. The (real) matrices  $\mathbf{R}_0$  and  $\mathbf{R}_{-1}$  are possibly not included in the model.

$$\mathbf{Y}^{n \times n}(\omega) = j\omega \mathbf{R}_{-1}^{n \times n} + \mathbf{R}_0^{n \times n} + \sum_{i=1}^N \frac{\mathbf{R}_i^{n \times n}}{j\omega - a_i} \quad (1)$$

$$\mathbf{i}^{n \times 1}(\omega) = \mathbf{Y}^{n \times n}(\omega) \cdot \mathbf{v}^{n \times 1}(\omega) \quad (2)$$

The objective is to calculate a minimal perturbation (3a) to the residue matrices  $\mathbf{R}_i$  such that the passivity condition (3b) is satisfied [16], i.e. the eigenvalues of  $\mathbf{G} = \text{Re}\{\mathbf{Y}\}$  are positive for all frequencies. (Positive eigenvalues of a general matrix  $\mathbf{A}$  is written “ $\mathbf{A} > 0$ ”).

$$\Delta \mathbf{Y}(\omega) \approx 0 \quad (3a)$$

$$\mathbf{G}(\omega) + \Delta \mathbf{G}(\omega) > 0 \quad \forall \omega \quad (3b)$$

Introducing (1) in (3a) and (3b) gives (4a) and (4b). Positive eigenvalues are also required for  $\mathbf{R}_0$  (4c) and  $\mathbf{R}_{-1}$  (4d).

$$j\omega \Delta \mathbf{R}_{-1} + \Delta \mathbf{R}_0 + \sum_{i=1}^N \frac{\Delta \mathbf{R}_i}{j\omega - a_i} \approx 0 \quad (4a)$$

$$\text{Re} \left\{ \mathbf{R}_0 + \Delta \mathbf{R}_0 + \sum_{i=1}^N \frac{\mathbf{R}_i + \Delta \mathbf{R}_i}{j\omega - a_i} \right\} > 0, \quad \forall \omega \quad (4b)$$

$$\mathbf{R}_0 > 0 \quad (4c)$$

$$\mathbf{R}_{-1} > 0 \quad (4d)$$

The same passivity condition also applies when the model (1) represents an impedance matrix  $\mathbf{Z}$ . In that case,  $\mathbf{Z}$  substitutes  $\mathbf{Y}$ .

To simplify the presentation, it is assumed that the model includes  $\mathbf{R}_0$  but not  $\mathbf{R}_{-1}$ . (The actual implementation also allows to include none or both terms).

## III. PASSIVITY ASSESSMENT AND ENFORCEMENT

The passivity enforcement is combined with a passivity assessment step as shown in Fig. 1. The passivity assessment identifies frequencies with maximum passivity violations using frequency sweeping [19], [25] or test matrices, [17], [18]. The frequency points are passed to the passivity enforcement step which calculates a model perturbation that attempts to remove the violations at these frequencies. The process of assessment and enforcement is repeated until all passivity violations have

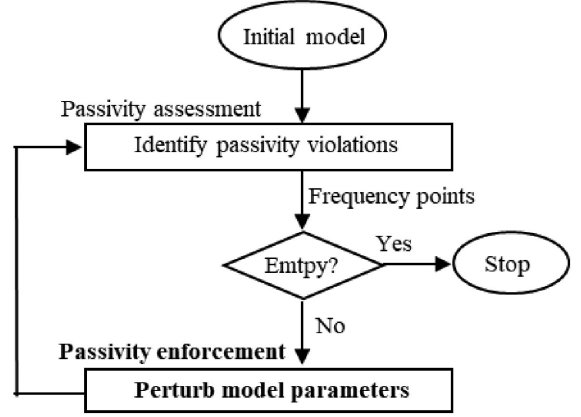


Fig. 1. Passivity assessment and enforcement.

been removed. The developments in this paper and presented efficiency metrics refer only to the passivity enforcement step.

## IV. RESIDUE PERTURBATION

This section briefly reviews the passivity enforcement scheme known as residue perturbation (RP) [19]. This scheme seeks to enforce the passivity condition (4b)-(4d) such that the change to the original model (4a) is minimized in the LS sense.

### A. Passivity Conforming Constraint

The relation between a matrix perturbation and the change to matrix eigenvalues is non-linear. By utilizing the eigenvalue perturbation lemma [26] one obtains a first-order linear approximation (5a) between a perturbation of  $\mathbf{G}$  and its  $j$ th eigenvalue, where  $\mathbf{v}$  and  $\mathbf{w}$  denote right (column) and left (row) eigenvector, respectively. This expression simplifies to (5b) when considering the symmetry and realness of  $\mathbf{G}$  and assuming the eigenvectors  $\mathbf{v}_j$  have been normalized to unit length.

$$\Delta \lambda_j \approx \frac{\mathbf{w}_j^T(\Delta \mathbf{G})\mathbf{v}_j}{\mathbf{w}_j^T \mathbf{v}_j} \quad (5a)$$

$$\Delta \lambda_j \approx \mathbf{v}_j^T(\Delta \mathbf{G})\mathbf{v}_j \quad (5b)$$

After writing (5b) for all eigenpairs and reordering the expression, one arrives [19] at the matrix-vector equation (6a) where matrix operator  $\text{vec}(\cdot)$  places all elements of  $\Delta \mathbf{G}$  in a column vector. Matrix  $\mathbf{P}$  is a sensitivity matrix of dimension  $n \times n^2$  that defines how the eigenvalues of  $\mathbf{G}$  change due to a (small) change in the elements of  $\mathbf{G}$ . One further introduces the (linear) relation (6b) between a perturbation of the residue matrices and the change to the elements of  $\mathbf{G}$ .  $\mathbf{S}$  is a sparse matrix with  $n^2(N+1)$  non-zero entries and  $\text{vec}(\Delta \mathbf{R})$  is a vector holding all  $n^2(N+1)$  elements of the residue matrices.

$$\Delta \lambda \approx \mathbf{P} \cdot \text{vec}(\Delta \mathbf{G}) \quad (6a)$$

$$\text{vec}(\Delta \mathbf{G}) = \mathbf{S} \cdot \text{vec}(\Delta \mathbf{R}) \quad (6b)$$

Eqs. (6a) and (6b) are combined to give the (linearized) relation (7) between  $\Delta \mathbf{R}$  and eigenvalues,  $\Delta \lambda$ .

$$\Delta \lambda \approx \mathbf{T} \cdot \text{vec}(\Delta \mathbf{R}), \quad \mathbf{T} = \mathbf{P}\mathbf{S} \quad (7)$$

### B. Constrained Linear Least Squares Problem

Eq. (4a) is expressed as a matrix-vector product (8) between  $\Delta \mathbf{R}$  and  $\Delta \mathbf{Y}$ , where  $\mathbf{M}$  is a sparse matrix with  $n^2(N+1)$  non-zero entries.

$$\mathbf{M} \cdot \text{vec}(\Delta \mathbf{R}) = \text{vec}(\Delta \mathbf{Y}) \approx \mathbf{0} \quad (8)$$

A minimal perturbation  $\text{vec}(\Delta \mathbf{Y})$  is obtained as the least squares solution of (9a) where  $\mathbf{x} = \text{vec}(\Delta \mathbf{R})$ . The (rectangular) system matrix  $\mathbf{A}$  is established by writing (8) for many frequency samples with  $\mathbf{M}$  constituting block rows of  $\mathbf{A}$ . Eq. (9a) is combined with the linear constraint equation (9b) where  $\mathbf{C}$  is built by writing (7) for frequency points with passivity violations, with matrix  $\mathbf{T}$  constituting block rows of  $\mathbf{C}$  and with the required eigenvalue shift  $\Delta \lambda$  constituting block rows of vector  $\mathbf{d}$ . Equations (9a) and (9b) are solved simultaneously as a constrained linear LS problem. It is remarked that a change of variable is used [10] such that  $\mathbf{x}$  and all matrices in (9) become real-valued.

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\| \quad (9a)$$

$$\mathbf{C}\mathbf{x} > \mathbf{d} \quad (9b)$$

### V. SOLVING VIA NON-NEGATIVE LEAST SQUARES (RP-NNLS)

The solving of (9) was in the original implementations of RP and FRP achieved using Quadratic Programming (QP). The calculation time can however become prohibitive for problems with many terminals and high orders. In order to overcome this problem, an alternative solving procedure [24] is introduced that is based on non-negative least squares (NNLS).

#### A. Conversion to Least Distance Problem (LDP)

The (rectangular) system matrix  $\mathbf{A}$  in (9a) is subjected to QR decomposition

$$\mathbf{A} = \bar{\mathbf{Q}}\bar{\mathbf{R}} \quad (10)$$

where  $\bar{\mathbf{Q}}$  is orthonormal and  $\bar{\mathbf{R}}$  is an upper triangular matrix. This gives for (9a)

$$\min_{\mathbf{x}} \|\bar{\mathbf{Q}}\bar{\mathbf{R}}\mathbf{x}\| \quad s.t. \quad \mathbf{C}\mathbf{x} > \mathbf{d} \quad (11)$$

Pre-multiplying with  $\bar{\mathbf{Q}}^T$  gives

$$\min_{\mathbf{x}} \|\bar{\mathbf{R}}\mathbf{x}\| \quad s.t. \quad \mathbf{C}\mathbf{x} > \mathbf{d} \quad (12)$$

Introducing a change of variable,  $\mathbf{y} = \bar{\mathbf{R}}\mathbf{x}$ , leads to a least distance problem (LDP),

$$\min_{\mathbf{y}} \|\mathbf{y}\| \quad s.t. \quad \mathbf{C}\bar{\mathbf{R}}^{-1}\mathbf{y} > \mathbf{d} \quad (13)$$

#### B. Non-Negative Least Squares (NNLS)

It is proved in [24] that by introducing matrices (14), problem (13) can be solved as the non-negative least squares (NNLS) problem (15), where  $\mathbf{y}$  and  $\mathbf{x}$  are afterwards recovered by (16) and (17) from the solution residual (18), with  $\text{col}_{\mathbf{E}}$  being the number of columns of  $\mathbf{E}$ . The dimension of  $\mathbf{E}$  is  $n^2(N+1) \times (L+1)$

where  $L$  is the number of constraints (rows in  $\mathbf{C}$ ).

$$\mathbf{E} = \begin{bmatrix} (\mathbf{C}\bar{\mathbf{R}}^{-1})^T \\ \mathbf{d}^T \end{bmatrix}, \quad \mathbf{f} = [0, \dots, 0 \ 1]^T \quad (14)$$

$$\min_{\mathbf{u}} \|\mathbf{E}\mathbf{u} - \mathbf{f}\| \quad s.t. \quad \mathbf{u} \geq \mathbf{0} \quad (15)$$

$$\mathbf{y} = -\frac{\varepsilon(1 : \text{col}_{\mathbf{E}} - 1)}{\varepsilon(\text{col}_{\mathbf{E}})} \quad (16)$$

$$\mathbf{x} = \bar{\mathbf{R}}^{-1}\mathbf{y} \quad (17)$$

$$\varepsilon = \mathbf{f} - \mathbf{E}\mathbf{u} \quad (18)$$

## VI. IMPLEMENTATION

### A. Block-Wise QR-Decomposition

The computation time of the RP-NNLS method is often dominated by the time needed for the QR decomposition in (10). This problem is overcome by building  $\mathbf{A}$  as a block-diagonal matrix where each block corresponds to one element of  $\mathbf{Y}$ ,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{n^2} \end{bmatrix} \quad (19)$$

Each matrix block has dimension  $2K \cdot (N+1)$  where  $K$  is the number of frequency samples. (The factor 2 results when formulating the system matrices using real-only numbers). For such matrix (with  $n^2$  blocks), the QR-decomposition can be applied to its individual blocks (20) since any two columns coming from different blocks are mutually orthogonal.

$$\mathbf{A} = \bar{\mathbf{Q}}\bar{\mathbf{R}} = \begin{bmatrix} \bar{\mathbf{Q}}_1\bar{\mathbf{R}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Q}}_2\bar{\mathbf{R}}_2 & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{Q}}_{n^2}\bar{\mathbf{R}}_{n^2} \end{bmatrix} \quad (20)$$

Equation (20) can be written

$$\mathbf{A} = \bar{\mathbf{Q}}\bar{\mathbf{R}} = \begin{bmatrix} \bar{\mathbf{Q}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Q}}_2 & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{Q}}_{n^2} \end{bmatrix} \cdot \begin{bmatrix} \bar{\mathbf{R}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{R}}_2 & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{R}}_{n^2} \end{bmatrix} \quad (21)$$

and therefore

$$\bar{\mathbf{R}}^{-1} = \begin{bmatrix} \bar{\mathbf{R}}_1^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{R}}_2^{-1} & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{R}}_{n^2}^{-1} \end{bmatrix} \quad (22)$$

From this observation it follows that the QR decomposition (10) can be applied to the individual blocks  $\mathbf{A}_j$  of  $\mathbf{A}$  independently, and that the back-transformation by (17) can be achieved

by applying individual matrices  $\bar{\mathbf{R}}_j^{-1}$  to the corresponding partitions of the solution vector  $\mathbf{y}$ ,

$$\mathbf{x}_j = \bar{\mathbf{R}}_j^{-1} \mathbf{y}_j \quad (23)$$

In the actual implementation, the matrix inverse in (23) is not performed as  $\mathbf{x}_j$  is calculated by solving  $\mathbf{y}_j = \mathbf{R}_j \mathbf{x}_j$  by a triangular solver via back substitution. Similarly, the matrix inverse is not used when calculating  $\mathbf{E}$  in (14) which involves the term  $\mathbf{C}\bar{\mathbf{R}}^{-1}$ . Instead, blocks  $\mathbf{C}_j \bar{\mathbf{R}}_j^{-1}$  are calculated from  $\mathbf{C}_j$  and  $\bar{\mathbf{R}}_j$  using Matlab's "slash" operator, again invoking solving by a triangular solver using back substitution.

The symmetry of  $\mathbf{Y}$  is utilized by including only the upper triangle of  $\mathbf{Y}$ , thereby reducing the number of blocks from  $n^2$  to  $n_{\text{tri}} = n(n+1)/2$ . A further improvement is achieved in the case that the LS frequency-dependent weighting is identical for all elements (e.g. unitary weighting). In that case, all blocks  $\mathbf{A}_j$  are identical and QR factorization needs only be performed for a single block.

### B. Solving the NNLS-Problem

The actual solving of the NNLS problem (15) is achieved by an algorithm described in [24], available in Matlab routine `lsqnonneg`. This is an active set method that solves (15) in a finite number of steps. The following outlines the algorithm, see [24] for full details.

The elements of the solution vector are split between an active set  $Z$  and an inactive set  $P$ . Elements in  $Z$  are kept at zero ( $\mathbf{u}_Z = \mathbf{0}$ ) while the elements of  $P$  ( $\mathbf{u}_P$ ) are free to vary. The calculation is initialized by setting  $\mathbf{u} = \mathbf{u}_Z$ ,  $P = \emptyset$ . This is a feasible solution since it satisfies  $\mathbf{u} \geq \mathbf{0}$ . An outer loop is entered where elements from  $Z$  are brought to  $P$ , one-by-one. In each iteration, the gradient vector (24) is calculated. The element in  $Z$  corresponding to the largest positive element of  $\mathbf{w}$  is moved from  $Z$  to  $P$ .

$$\mathbf{w} = \mathbf{E}^T (\mathbf{f} - \mathbf{E}\mathbf{u}_Z) \quad (24)$$

A sub-vector  $\mathbf{u}_P$  of  $\mathbf{u}$  is calculated as the (unconstrained) LS solution of

$$\mathbf{E}_P \mathbf{u}_P = \mathbf{f} \quad (25)$$

where  $\mathbf{E}_P$  contains the corresponding columns of  $\mathbf{E}$ . (The remaining elements in  $\mathbf{u}$  are kept at zero value in sub-vector  $\mathbf{u}_Z$ ). The solving of (25) may result in that some elements of  $\mathbf{u}_P$  become negative. In that case, an inner loop is entered which refines the elements of  $\mathbf{u}_P$  such that the constraints are satisfied, followed by an update of sets  $P$  and  $Z$ . The main loop involving (24) and (25) is repeated until the set  $Z$  is empty or all elements in  $\mathbf{w}$  are non-positive.

## VII. COMPUTATIONAL ANALYSIS

The scalability of the method is studied next. Consider a single passivity enforcement step of a pole-residue model with  $n$  terminals and  $N$  pole-residue terms by solving the constrained LS problem (9) with  $K$  frequency samples and  $L$  passivity violations. To simplify the analysis, it is assumed that symmetry of  $\mathbf{Y}$  is not exploited, i.e. all  $n^2$  elements are included in the

TABLE I  
APPROXIMATE FLOP COUNTS FOR BASIC MATRIX OPERATIONS

Operation	Flops
Eigenvalue decomposition of $n \times n$ matrix [27]	$\frac{4}{3}n^3$
QR decomposition of $m \times n$ matrix [27]	$\sim 2mn^2$
Multiplying $m \times n$ matrix with $n \times n$ matrix	$\sim 2mn^2$
Solving $\mathbf{y}=\mathbf{R}\mathbf{x}$ with $\mathbf{R}$ upper triangular [27]	$\sim n^2$

system equations. Also, it is assumed that  $\mathbf{R}_0$  and  $\mathbf{R}_{-1}$  are not present in the model. The number of floating-point operations (flops) for basic matrix operations are listed in Table I, with  $m$  and  $n$  being general matrix dimensions.

### A. Forming LS Equation

Each of the  $n^2$  blocks  $\mathbf{A}_j$  is of dimension  $2K \times N$ . The building of each block requires  $k_1 KN$  flops while the QR decomposition of each  $\mathbf{A}_j$  requires  $\sim 4KN^2$  flops, giving the total count in (26a). In the case of common weighting, only a single block needs to be built and subjected to QR decomposition, giving the reduced flop count in (26b).

$$C_1 \sim k_1 K n^2 N + 4K n^2 N^2 \quad (26a)$$

$$C_1 \sim k_1 KN + 4KN^2 \quad (26b)$$

### B. Forming Constraint Equation

The constraint matrix  $\mathbf{C}$  in (9b) is built via matrix  $\mathbf{P}$  (6a) and the sparse  $\mathbf{S}$  (6b).

The forming of  $\mathbf{P}$  requires to perform the eigenvalue decomposition of matrix  $\mathbf{G}(\omega_j)$  of dimension  $n \times n$  for all  $K_{\text{viol}}$  frequency points with violation(s), and to calculate sensitivities for violating eigenvalues via (5). The flop count is about

$$C_2 \sim \frac{4}{3} K_{\text{viol}} n^3 + 2L n^2 \quad (27)$$

The forming of  $\mathbf{T}$  in (7) requires to multiply each element of  $\mathbf{P}$  of dimension  $L \times n^2$  with  $N$  elements from the sparse matrix  $\mathbf{S}$ , requiring  $L n^2 N$  multiplications. The flop count is

$$C_3 \sim 2L n^2 N \quad (28)$$

### C. Forming NNLS Problem

The constraint equation  $\mathbf{C}$  is converted into matrix  $\mathbf{E}$  from  $n^2$  partitions  $\mathbf{C}_j$  of dimension  $L \times N$  and submatrices  $\bar{\mathbf{R}}_j^{-1}$  of dimension  $N \times N$  by calculating terms  $\mathbf{C}_j \bar{\mathbf{R}}_j^{-1}$ . The calculation time is about

$$C_4 \sim L n^2 N^2 \quad (29)$$

### D. Solving NNLS Problem

The computation time of NNLS is dominated by the solving of (25). The computational efficiency of Matlab routine `lsqnonneg` was improved by solving (25) using Normal Equations (30), i.e.

$$\mathbf{u}_P := (\mathbf{E}_P^T \mathbf{E}_P)^{-1} \mathbf{E}_P^T \mathbf{f} \quad (30)$$

It is observed that the matrix to be inverted in (30) is equal to the number of elements in set  $P$ , i.e. the number of inactive constraints. The solving is performed repeatedly with the number of columns increasing by one in each iteration. In the actual implementation, the calculation time is reduced as follows. At the initialization,  $\mathbf{E}$  is calculated with all columns included,

$$\mathbf{F} = \mathbf{E}^T \mathbf{E} \quad (31)$$

In the iterations, the matrix  $\mathbf{E}_P^T \mathbf{E}_P$  in (25) is extracted as a subset of  $\mathbf{F}$ . The computational complexity of forming  $\mathbf{F}$  is

$$C_5 \sim 2Ln^2N \quad (32)$$

During the repeated solving within NNLS, the matrix to be inverted in (30) grows from unity ( $p = 1$ ) up to the value  $p_{\max}$  where all constraints are satisfied. The computational cost is given as a finite sum of powers (33) whose total is analytical [28].

$$C_6 \sim k_5 \sum_{p=1}^{p_{\max}} p^3 = k_5 \frac{1}{4} (p_{\max}^4 + 2p_{\max}^3 + p_{\max}^2) \quad (33)$$

In a worst case scenario, one could end up with a single active constraint and one needs to invert a matrix which increases in dimension,  $p = 1, 2 \dots p_{\max} = L$ . In the implemented code, the number of constraints  $L$  (columns of  $\mathbf{E}$ ) is made small or moderate by selecting only violating eigenvalues when building  $\mathbf{C}$  and  $\mathbf{d}$ . As a result, most constraints in NNLS will be active, thus limiting the number of inactive constraints, i.e. the maximum value of  $p$ .

An additional cost results from the repeated calculation of the gradient by the two matrix-vector multiplications in (24). The cost of a single matrix-vector multiplication involving  $\mathbf{E}$  is

$$C_7 \sim 2Ln^2N \quad (34)$$

### VIII. IMPROVING COMPUTATIONAL EFFICIENCY

The block-wise RP-NNLS method has been implemented with two options that can further reduce computation time.

#### A. Matrix Bandwidth Reduction

Following the procedure in [20], the user can limit the number of elements in  $\mathbf{Y}$  to be perturbed by specifying the matrix bandwidth via parameter  $bw$ , defined by (35). For instance, specifying  $bw = 0$  will only use diagonal elements while specifying  $bw = n - 1$  will use all elements. As a result, the number of columns in  $\mathbf{A}$  and  $\mathbf{C}$  and thus rows in  $\mathbf{E}$  is reduced accordingly. This approach is useful for problems with a high number of terminals as will be demonstrated in Section X.B.

$$\begin{bmatrix} 0 & 1 & 2 & \cdots & n-1 \\ 1 & 0 & 1 & 2 & \vdots \\ 2 & 1 & \ddots & 1 & 2 \\ \vdots & 2 & 1 & 0 & 1 \\ n-1 & \cdots & 2 & 1 & 0 \end{bmatrix} \quad (35)$$

#### B. Pole-Residue Term Reduction

The user can in addition specify an array of indices to pole residue terms that are to be perturbed [20]. This approach is useful for problems with a high number of pole-residue terms. The pole-residue terms can for instance be selected in frequency areas where passivity violations occur, e.g. by an automated selection scheme as in [9]. Again, the number of columns in  $\mathbf{A}$  and  $\mathbf{C}$ , and thus rows in  $\mathbf{E}$ , becomes reduced.

#### C. Adjusted Step Length

The constraint equation is based on linearized eigenvalue perturbation by (5b). Therefore, the solving of the passivity enforcement step will in general not bring a violating eigenvalue exactly to the zero line, often requiring additional iterations. The problem is mitigated by slightly increasing the correction step length, i.e. by replacing (9) with (36) where  $\alpha \geq 1$ . A threshold value  $\beta$  is also available, see Section XI.C for details.

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\| \text{ s.t. } \mathbf{C}\mathbf{x} > \alpha \cdot \mathbf{d} + \beta \quad (36)$$

#### D. Single Building of Objective Function

The LS part (9a) is kept unchanged during the iterations, similarly as in the previous works [19], [21]. Therefore, the block(s)  $\mathbf{A}_j$  and its associated QR-decomposition is calculated a single time, giving a further reduction of the total computation time.

### IX. MEMORY REQUIREMENTS

The blocks  $\mathbf{A}_j$  of  $\mathbf{A}$  of (19) are built one at a time and subjected to QR-decomposition. Since only a single block needs to exist on the computer, the storage requirements are that of a single matrix  $\mathbf{A}_j$  of dimension  $2K \times N$ . The QR decomposition gives  $n_{\text{tri}} = n(n+1)/2$   $\bar{\mathbf{R}}_j$  matrices that need to be stored for later use in (23), requiring a total memory of  $n_{\text{tri}}N^2$ . In the case that the LS weighting is identical for all elements of  $\mathbf{Y}$  (e.g. unitary weighting), all blocks  $\mathbf{A}_j$  are identical and so a single (small) QR-decomposition needs to be calculated and only a single  $\bar{\mathbf{R}}_j$  matrix of dimension  $N \times N$  needs to be stored. The largest matrix that needs to be built is  $\mathbf{E}$  in (14) which has dimension  $(n_{\text{tri}} + 1)N \times (L + 1)$ . That matrix is built sequentially from partitions of  $\mathbf{P}$  and  $\mathbf{S}$ , and those matrices need therefore not be built in full size. The number of constraints  $L$  is kept small by establishing frequency bands of passivity violations. In each band, only the maximum violation (of each violating eigenvalue) is included as constraint [21].

### X. EXAMPLES

Three examples are presented which demonstrate the performance of the RP-NNLS passivity enforcement scheme in Matlab version R2018a. The acceleration factor  $\alpha$  in (36) is 1.05, similarly as in [9]. The listed CPU times are for the passivity enforcement steps, excluding passivity assessment (see Fig. 1). All calculations start with a “clear all” statement in Matlab, except for the CPU timing plots in Figs 4 and 9. All calculations

TABLE II  
COMPARISON OF PASSIVITY ENFORCEMENT TECHNIQUES FOR DIFFERENT  
FDNE MODEL ORDERS. RMS-ERROR

$n$	$N$	RMS-error	
		RP-NNLS	RP-QP
15	90	$2.8 \times 10^{-6}$	$2.8 \times 10^{-6}$
15	95	$2.8 \times 10^{-7}$	$3.4 \times 10^{-7}$
15	100	$3.3 \times 10^{-6}$	$3.3 \times 10^{-6}$
15	105	$7.9 \times 10^{-6}$	$8.0 \times 10^{-6}$
15	150	$2.5 \times 10^{-5}$	$2.4 \times 10^{-5}$

TABLE III  
COMPARISON OF PASSIVITY ENFORCEMENT TECHNIQUES FOR DIFFERENT  
FDNE MODEL ORDERS. CPU TIME

$n$	$N$	CPU [sec]		
		Total	NNLS alone	QP alone
15	90	0.10	<b>0.014</b>	<b>1.08</b>
15	95	0.11	<b>0.014</b>	<b>1.23</b>
15	100	0.10	<b>0.016</b>	<b>1.26</b>
15	105	0.11	<b>0.015</b>	<b>1.27</b>
15	150	0.27	<b>0.050</b>	<b>9.27</b>

are performed on a 64-bit Windows 10 laptop with 16 GB RAM and an Intel i7-7600U @2.8 GHz CPU.

#### A. FDNE Modeling

This example considers the passivity enforcement of a frequency-dependent network equivalent (FDNE) with  $K = 751$  frequency samples and unitary LS weighting. The example data set is the same as used in [9]. The useful frequency band is from 1 Hz to 1500 Hz, making the model suitable for simulation of low-frequency transients. The initial model to be passivated has five three-phase terminals ( $n = 15$ ) and five alternative model orders ( $N$ ) are considered. The passivity enforcement considers all matrix elements ( $bw = 14$ ) and all pole-residue terms are included. Passivity is enforced in two alternative ways: using RP-NNLS or using the original RP formulation [19] with QP as solver. The QP solution algorithm is an interior-point-convex method implemented in Matlab routine quadprog.

Table II shows that the resulting RMS-error by the two methods is practically the same. Both methods enforce passivity in a single iteration, except for the last case ( $N = 150$ ) where RP-NNLS and RP-QP need three iterations.

Table III shows the resulting CPU times. With RP-NNLS, the total time (building matrices and solving NNLS) is listed, as well as the time for solving NNLS. With RP-QP, only the time used for solving the QP problem is listed (“QP alone”). It is clearly seen that solving the NNLS problem is much faster than solving the QP problem.

Fig. 2 shows the impact of the passivity enforcement on the eigenvalues of  $\mathbf{G}(\omega)$  with  $N = 150$ , with usage of RP-NNLS. It is observed that the passivity violations have been successfully removed, i.e. all eigenvalues are positive. Fig. 3 shows the impact

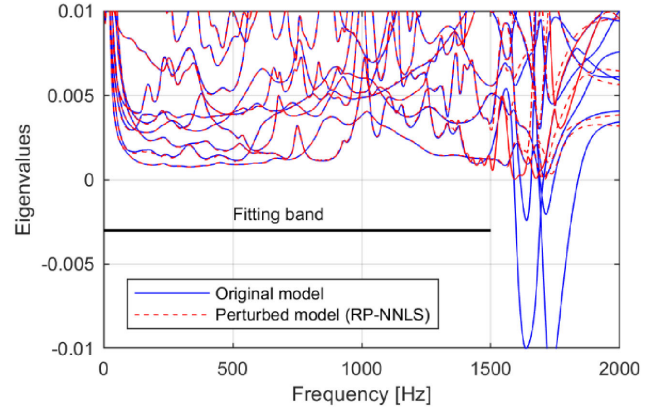


Fig. 2. Perturbed eigenvalues of  $\mathbf{G}$  ( $n = 15$ ,  $N = 150$ ).

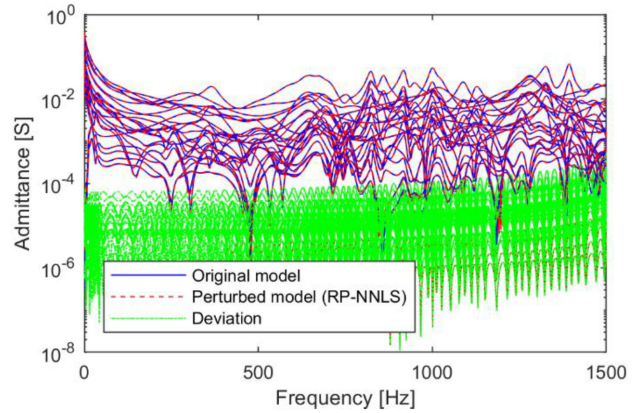


Fig. 3. Perturbed elements of  $\mathbf{Y}$  ( $n = 15$ ,  $N = 150$ ).

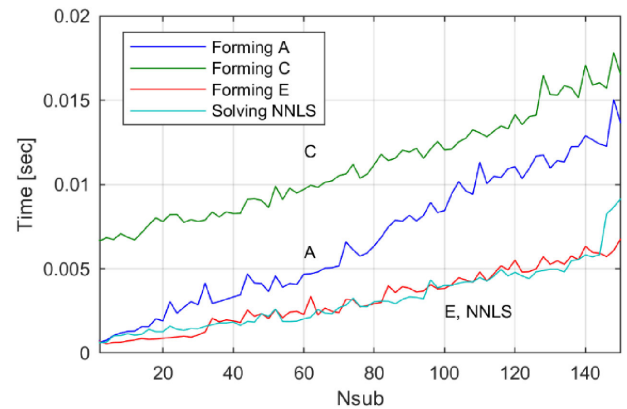


Fig. 4. CPU time vs. number of pole-residue terms used in passivity enforcement. First iteration result.

on the model’s admittance elements. The perturbation is seen to be very small inside the fitting band, relative to the magnitude of the admittance elements.

Fig. 4. shows the CPU time as function of the number of pole-residue (PR) terms ( $N_{\text{sub}}$ ) included in the passivity enforcement step (Section VIII.B), for the case with  $N = 150$ . The CPU time is for the first iteration (out of 3), calculated as the average of 50 runs. With a low number of PR terms, the total

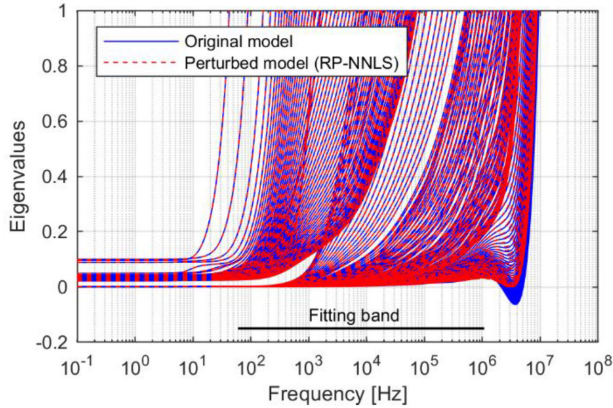


Fig. 5. Perturbed eigenvalues of  $\text{Re}\{\mathbf{Z}_b\}$  ( $n = 213, N = 10$ ).

calculation time is dominated by the calculation of the constraint matrix  $\mathbf{C}$ . That matrix has a substantial computational effort that is independent on  $N_{\text{sub}}$ : calculation of sample values for  $\mathbf{G}$ , eigenvalue decomposition of  $\mathbf{G}$ , and calculation of eigenvalue sensitivities (5b). In addition comes one term (28) which is linear in  $N_{\text{sub}}$ .

The CPU cost for forming  $\mathbf{A}$  is superlinear as can be expected from (26b) which includes linear and quadratic terms in  $N$ . (Only a single block  $\mathbf{A}_j$  is calculated and subjected to QR decomposition since unitary weighting is used).

The CPU cost for forming  $\mathbf{E}$  and solving NNLS is weakly superlinear in  $N_{\text{sub}}$  while a more quadratic growth was expected. This discrepancy can in part be attributed to the time needed for accessing and writing data to arrays, as well as overhead in the computations.

### B. White-Box Transformer Branch Impedance Modeling

This example considers frequency-dependent branch impedance modeling as part of white-box (detailed) transformer modeling [14]. The transformer is a single-phase 50 MVA three-winding transformer with rated voltage 230/69/13.8 kV at 60 Hz. The branch impedance matrix  $\mathbf{Z}_b$  has been calculated as function of frequency using FEM. The data set is a matrix of dimension  $213 \times 213$ , given as 14 logarithmically spaced frequency samples between 60 Hz and 1.08 MHz. The matrix  $\mathbf{Z}_b$  is fitted with a pole-residue model of order  $N = 10$ , with inclusion of  $\mathbf{R}_0$ . The resulting model has passivity violations at out-of-band frequencies, at both low and high frequencies.

Passivity is enforced using RP-NNLS with  $K = 14$  frequency samples and perturbation of all residue matrices and all matrix elements, and with individual LS weighting for all impedance elements  $(i, j)$ ,

$$\text{weight}_{i,j}(\omega) = \frac{1}{\sqrt{Z_b(i,j)(\omega)}} \quad (37)$$

Figs. 5 and 6 (zoomed view) show the impact of passivity enforcement on the eigenvalues of  $\text{Re}\{\mathbf{Z}_b(\omega)\}$  while Fig. 7 shows the impact on the last column of  $\mathbf{Z}_b$  within the fitting band. It is observed that passivity is successfully enforced as all eigenvalues are positive. The traces of the perturbed model

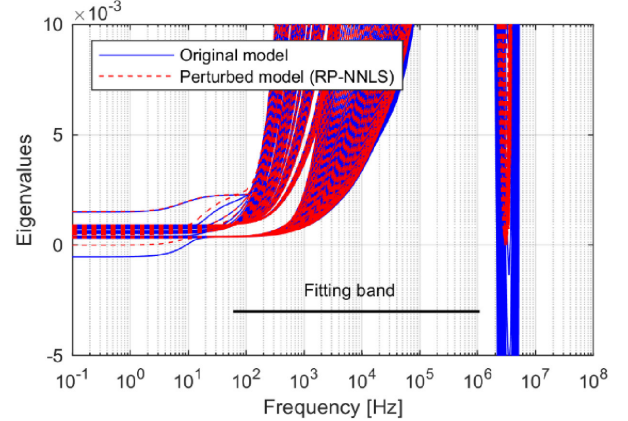


Fig. 6. Zoomed view of Fig. 5.

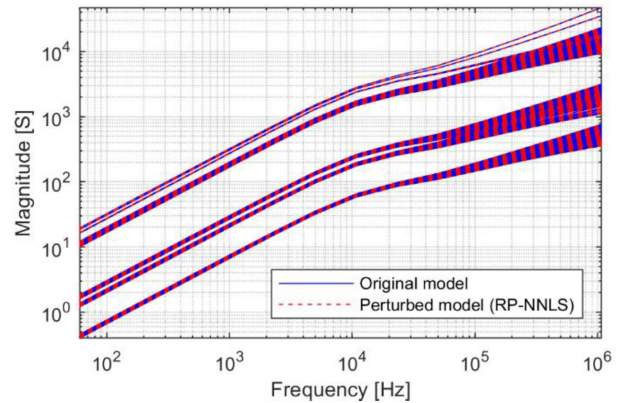


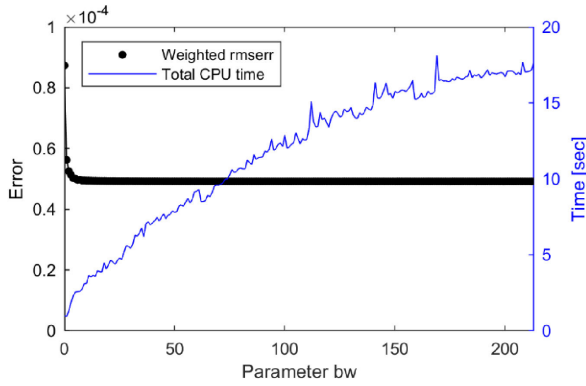
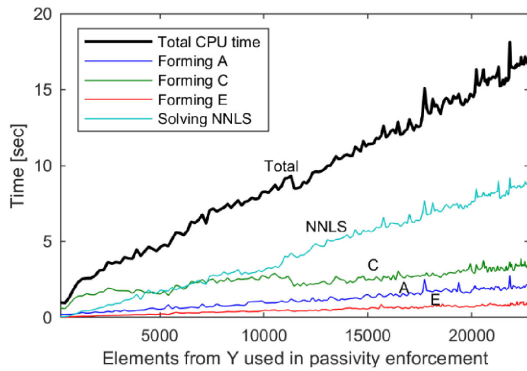
Fig. 7. Last column of perturbed  $\mathbf{Z}_b$  of case #1 ( $n = 213, N = 10$ ).

TABLE IV  
PASSIVITY ENFORCEMENT OF TRANSFORMER BRANCH IMPEDANCE MATRIX CASE

Case	$n$	$bw$	PR terms	Iterations	RMS-error (weighted)	CPU [s]
1	213	213	10	3	$4.9 \times 10^{-5}$	<b>16.0</b>
2	213	213	2	3	$1.7 \times 10^{-3}$	<b>3.1</b>
3	213	0	10	3	$8.7 \times 10^{-5}$	<b>0.9</b>
4	213	0	2	3	$3.1 \times 10^{-3}$	<b>0.7</b>

in Fig. 7 are virtually overlapping those of the original model, implying a very small perturbation.

Table IV lists the weighted RMS-error and CPU time for the calculation (average of 5 runs). Case 1 is for the above calculation which uses all elements and all residue matrices as free variables. Passivity is enforced in 16.0 sec with 3 iterations. The next rows (Cases 2-4) show the result when reducing the number of free variables. The reduction is achieved in two ways: 1) using only the diagonal elements in each residue matrix ( $bw = 0$ , see Section VIII.A), and 2) including only the first and last pole-residue term (Section VIII.B). The calculation time is reduced to 0.9 sec with  $bw = 0$ . Usage of fewer pole-residue terms gives in this example a substantial increase of the fitting error. One test was also performed where the number of samples

Fig. 8. Weighted RMS-error and CPU time vs. parameter  $bw$ .Fig. 9. CPU time vs. number of perturbed elements in  $Y$ .

$K$  was increased from 14 to 201. With Case 1, four iterations were needed, and the total CPU time increased from 16.0 sec to 32 sec.

Fig. 8 shows the (weighted) RMS-error and CPU time as function of the parameter  $bw$ , when including all pole-residue terms. It is seen that usage of any low value of  $bw$  above zero achieves a fast calculation with only a small increase of the perturbation error.

Fig. 9 shows the individual contributions to the total CPU time when increasing  $bw$  from 0 to  $n - 1 = 212$  with use of all pole-residue terms, plotted as function of the total number of residue matrix elements that are perturbed. The contributions are seen to increase about linearly with the number of free variables. This result is as expected since most of the theoretical CPU time contributions in Section VII are proportional to  $n^2$ , i.e. proportional to the number of elements. The CPU time for building of the constraint matrix  $C$  is however seen to be nearly constant. That matrix has a substantial computational effort that is independent of  $bw$ : calculation of  $G$ , its eigenvalue decomposition, and the eigenvalue sensitivities (5b). The number of iterations in Fig. 9 varied between 3 and 6, staying constant at 3 above 11315 elements ( $bw > 61$ ).

With usage of RP-QP, the calculation time is excessive with the usage of the inverse weighting scheme (37). To see this, passivity is enforced for a sub-part of the model,

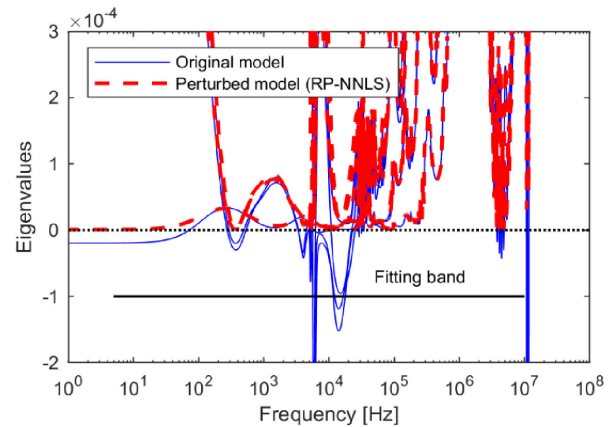
$$\mathbf{R}_{\text{sub},j} = \mathbf{R}_j(1 : n_{\text{sub}}, 1 : n_{\text{sub}}), j = 0, 1, \dots, N \quad (38)$$

TABLE V  
COMPARISON OF PASSIVITY ENFORCEMENT TECHNIQUES FOR ALTERNATIVE SUB-PARTS OF MODEL. CPU TIME

	CPU [sec]		
	RP-NNLS		RP-QP
$n_{\text{sub}}$	Total	NNLS alone	QP alone
10	0.22	<b>0.03</b>	<b>0.30</b>
20	0.24	<b>0.04</b>	<b>2.24</b>
30	0.37	<b>0.05</b>	<b>13.3</b>
40	0.42	<b>0.08</b>	<b>62.2</b>

TABLE VI  
PASSIVITY ENFORCEMENT OF BLACK-BOX TRANSFORMER MODEL

$n$	$N$	RP-NNLS		
		$N_{\text{iter}}$	RMS-error	CPU [s]
9	120	4	$1.5 \times 10^{-5}$	<b>0.28</b>

Fig. 10. Perturbed eigenvalues of  $G$  ( $n = 9, N = 120$ )

Passivity is enforced using all elements ( $bw = n_{\text{sub}} - 1$ ) and all  $N = 10$  pole-residue terms. Table V lists the CPU time for enforcing passivity, with alternative of values  $n_{\text{sub}}$ . It is observed that RP-NNLS offers superior computational performance.

### C. Black-Box Transformer Modeling

This example considers the modeling of a DFIG wind turbine transformer from frequency sweep admittance measurements. The transformer is a 33/6/0.66 kV Dynyn unit with both neutral points grounded. The admittance matrix has  $n = 9$  terminals and has been measured at  $K = 401$  logarithmically spaced frequency samples between 5 Hz and 10 MHz. The resulting model extracted by vector fitting ( $N = 120$ ) has several in-band and out-of-band passivity violations.

Table VI lists the RMS-error and CPU time for the passivity enforcement, using RP-NNLS with  $K = 401$  frequency samples and unitary weighting. Passivity is enforced in 0.28 sec. Fig. 10 shows the impact of the passivity enforcement on the eigenvalues of  $G$ . All passivity violations are removed, resulting in positive eigenvalues at all frequencies.



## XI. DISCUSSION

### A. Frequency Samples Selection

Although the frequency samples for the LS part (9a) can be freely chosen, one will normally use the same samples as was used in the original model extraction by vector fitting (VF). This sample selection permits to correct quite large out-of-band passivity violations without corrupting the in-band model behavior since the out-of band frequencies are associated with zero weight. A few additional frequency samples are added at out-of band frequencies (with a low LS weight) to improve the conditioning of (9a) [21].

### B. LS Weighting and Accuracy

It is practical to use the same LS weighting strategy as was used in VF when extracting the original model. That way, one seeks to retain the accuracy at the same frequencies and elements. Typical examples are unitary weighting and inverse magnitude weighting for relative error control.

One must however be aware that accuracy in the frequency domain does not translate into accuracy in the time domain. In particular, the errors of an admittance model can be largely magnified when the model is applied in a simulation with high-impedance terminations. In the extreme case with currents applied to all terminals, the terminal voltages become the response to the current excitation. In this impedance case, the small eigenvalues of  $\mathbf{Y}(\omega)$  become the large eigenvalues of  $\mathbf{Z}(\omega)$  [29]. Therefore, if the fitting of the original  $\mathbf{Y}$  and the subsequent passivity enforcement of the model gives a substantial relative error in the small eigenvalues of  $\mathbf{Y}$ , very large errors will result in the elements of  $\mathbf{Z}$  and therefore also in a time domain simulation. Special versions of VF (modal VF [29]) and RP (modal RP) [30] were introduced to mitigate such situations, as well as the use of similarity transformations [31].

### C. Passivity Enforcement Tolerance

The constraint equation (36) makes use of an adjusted step length by a factor  $\alpha \geq 1$  and a threshold value  $\beta \geq 0$ . A nonzero threshold value implies that the passivity enforcement makes the model slightly non-passive at frequencies with passivity violations. The intention is the possibility of giving damping to undamped oscillations.

The implemented code also includes a tolerance parameter  $\beta_{\mathbf{R}_0} \geq 0$  which acts on  $\mathbf{R}_0$  in (4c). In the case of an  $\mathbf{R}_0$  having one or more negative eigenvalues, this parameter prevents the passivity enforcement to give a matrix  $\mathbf{R}_0$  with one or more zero eigenvalues (singular matrix). Non-singularity of  $\mathbf{R}_0$  is a requirement when calculating bands of passivity violations using test matrices [18].

### D. Inner Loop Iterations

In some situations, with usage of strong weighting schemes (e.g. inverse magnitude weighting), the passivity enforcement can cause new passivity violations to arise. In such situation, the

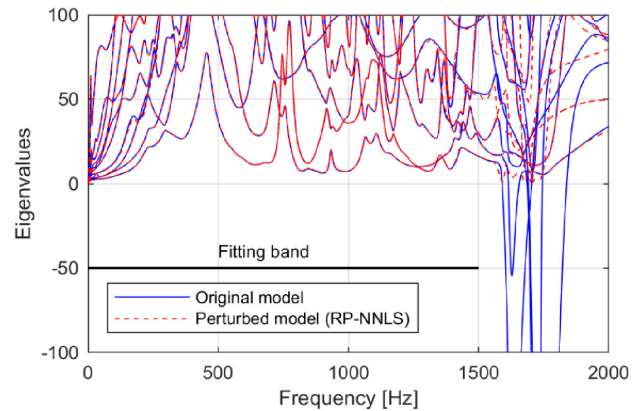


Fig. 11. Eigenvalues of  $\text{Re}\{\mathbf{Z}\}$  for FDNE example in Section X.A.

passivity scheme in Fig. 1 can be complemented with an inner loop that adds additional constraints, e.g. as shown in [21]. That approach has been included in the RP-NNLS implementation but was not necessary to use in the presented examples.

### E. Passivity Checking Using $\mathbf{Z}$

The passivity checking and enforcement based on  $\mathbf{Y}$  implies that the final model behaves as a passive component under any terminal condition. To verify this, consider the extreme case with current excitation on all terminals such that the voltage response is given by the model's impedance behavior. Recall from Section II that passivity of an impedance matrix  $\mathbf{Z}$  implies that the real part of  $\mathbf{Z}$  has positive eigenvalues for all frequencies. Fig. 11 shows the eigenvalues of  $\text{Re}\{\mathbf{Z}\}$  for the FDNE example in Section X.A, where  $\mathbf{Z}$  has been calculated from the model's admittance matrix,  $\mathbf{Z}(\omega) = \mathbf{Y}^{-1}(\omega)$ , before and after passivity enforcement. It is observed that the passivity enforcement of the model's  $\mathbf{Y}$  results in that  $\text{Re}\{\mathbf{Z}\}$  gets positive eigenvalues, as expected. Comparison with Fig. 2 further shows that the band with passivity violations of  $\mathbf{Z}$  is identical to that of  $\mathbf{Y}$ , ranging from 1590 Hz to 1838 Hz.

### F. Final Remarks

Further work in this area is still needed, for instance to reduce the linearization errors associated with the passivity constraint equation. Also, the possibility of inaccurate or unstable time domain simulation results should be subjected to more research.

## XII. CONCLUSION

A new implementation of the residue perturbation (RP) method is presented. It is based on a conversion of the constrained least squares problem to a least distance problem (LDP) that is solved as a non-negative least square (NNLS) problem. With multi-terminal models, the inherent sparsity of the system matrix is fully exploited, allowing the required QR-decomposition to be calculated independently for many small sub-matrices rather than for a single large matrix. This feature makes the method suitable for problems with many terminals.

The resulting NNLS problem to be solved is characterized by a system matrix having few columns and is efficiently solved using an active set algorithm.

The method was successfully applied for passivity enforcement to challenging examples: FDNE modeling with many terminals ( $n = 15$ ) and high orders ( $N = 90$ -150), transformer white-box impedance branch modeling with a high number of terminals ( $n = 213$ ) but low order ( $N = 10$ ), and measurement-based black-box transformer modeling ( $n = 9$ ,  $N = 120$ ).

The use of this efficient passivity enforcement scheme opens for more widespread application of rational modeling techniques in power systems.

#### ACKNOWLEDGEMENT

The author would like to thank Abner Ramirez (CINVESTAV, Guadalajara, Mexico) and other authors of [9] for permission to use their FDNE test cases in Section X.A, and Luiz Oliveira (WEG Transformers, Blumenau, Brazil), Alvaro Portillo (Montevideo, Uruguay) and Federico Portillo (Montevideo, Uruguay) for providing the transformer branch impedance matrix in Section X.B.

#### REFERENCES

- [1] J. R. Marti, "Accurate modelling of frequency-dependent transmission lines in electromagnetic transient simulations," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 1, pp. 147–157, Jan. 1982.
- [2] A. Morched, B. Gustavsen, and M. Tartibi, "A universal model for accurate calculation of electromagnetic transients on overhead lines and underground cables," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1032–1038, Jul. 1999.
- [3] M. Cervantes, I. Kocar, J. Mahseredjian, and A. Ramirez, "Accurate and reduced order identification of propagation function for electromagnetic transient analysis of cables," *IEEE Trans. Power Del.*, vol. 35, no. 2, pp. 968–976, Apr. 2020.
- [4] A. S. Morched, J. H. Ottevangers, and L. Marti, "Multi-port frequency dependent network equivalents for the EMTP," *IEEE Trans. Power Del.*, vol. 8, no. 3, pp. 1402–1412, Jul. 1993.
- [5] M. Abdel-Rahman, A. Semlyen, and M. Reza Iravani, "Two-layer network equivalent for electromagnetic transients," *IEEE Trans. Power Del.*, vol. 18, no. 4, pp. 1328–1335, Oct. 2003.
- [6] X. Lin, A. M. Gole, and M. Yu, "A wide-band multi-port system equivalent for real-time digital power system simulators," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 237–249, Feb. 2009.
- [7] U. D. Annakkage *et al.*, "Dynamic system equivalents: A survey of available techniques," *IEEE Trans. Power Del.*, vol. 27, no. 1, pp. 411–420, Jan. 2012.
- [8] D. N. Hussein, M. Matar, and R. Iravani, "A wideband equivalent model of type-3 wind power plants for EMT studies," *IEEE Trans. Power Del.*, vol. 31, no. 5, pp. 2322–2331, Oct. 2016.
- [9] J. Morales, J. Mahseredjian, K. Sheshyekani, A. Ramirez, E. Medina, and I. Kocar, "Pole-selective residue perturbation technique for passivity enforcement of FDNEs," *IEEE Trans. Power Del.*, vol. 33, no. 6, pp. 2746–2754, Dec. 2018.
- [10] 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.
- [11] S. Lefteriu and A. C. Antoulas, "A new approach to modeling multiport systems from frequency-domain data," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 29, no. 1, pp. 14–27, Jan. 2010.
- [12] J. Morales-Rodriguez, E. Medina, J. Mahseredjian, A. Ramirez, K. Sheshyekani, and I. Kocar, "Frequency-domain fitting techniques: A review," *IEEE Trans. Power Del.*, vol. 35, no. 3, pp. 1102–1110, Jun. 2020.
- [13] B. Gustavsen, "Wide band modeling of power transformers," *IEEE Trans. Power Del.*, vol. 19, no. 1, pp. 414–422, Jan. 2004.
- [14] E. E. Mombello and K. Moller, "New power transformer model for the calculation of electromagnetic resonant transient phenomena including frequency-dependent losses," *IEEE Trans. Power Del.*, vol. 15, no. 1, pp. 167–174, Jan. 2000.
- [15] M. Popov, L. Grcev, H. K. Høidalen, B. Gustavsen, and V. Terzija, "Investigation of the overvoltage and fast transient phenomena on transformer terminals by taking into account the grounding effects," *IEEE Trans. Industry Appl.*, vol. 51, no. 6, pp. 5218–5227, Nov./Dec. 2015.
- [16] S. Boyd and L.O. Chua, "On the passivity criterion for LTI n-ports," *Circuit Theory Appl.*, vol. 10, pp. 323–333, 1982.
- [17] S. Grivet-Talocia, "Passivity enforcement via perturbation of Hamiltonian matrices," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 9, pp. 1755–1769, Sep. 2004.
- [18] A. Semlyen and B. Gustavsen, "A half-size test matrix for fast and reliable assessment of rational models," *IEEE Trans. Power Del.*, vol. 24, no. 1, pp. 345–351, Jan. 2009.
- [19] B. Gustavsen and A. Semlyen, "Enforcing passivity for admittance matrices approximated by rational functions," *IEEE Trans. Power Syst.*, vol. 16, pp. 97–104, Feb. 2001.
- [20] B. Gustavsen, "Computer code for passivity enforcement of rational macromodels by residue perturbation," *IEEE Trans. Adv. Pack.*, vol. 30, no. 2, pp. 209–215, May 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] E. Medina, A. Ramirez, J. Morales, and J. Mahseredjian, "Alternative approach to alleviate passivity violations of rational-based fitted functions," *IEEE Trans. Power Del.*, vol. 34, no. 3, pp. 1161–1170, Jun. 2019.
- [23] E. Medina, A. Ramirez, J. Morales, and K. Sheshyekani, "Passivity enforcement of FDNEs via perturbation of singularity test matrix," *IEEE Trans. Power Del.*, vol. 35, no. 4, pp. 1648–1655, Aug. 2020.
- [24] C. L. Lawson and R.J. Hanson, "Solving least squares problems," SIAM, 1995. ISBN 0-89871-356-0.
- [25] 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.
- [26] G. H. Golub and C. V. Van Loan, *Matrix Computations*. New York, NY, USA: The John Hopkins Univ. Press, 1989, ISBN 08018-3739-1.
- [27] L.N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, 1997. ISBN 0-89871-361-7.
- [28] D. Zwillinger Editor, *Table of Integrals, Series and Products*. San Diego, CA, USA: Academic Press, 2015. ISBN 978-0-12-384933-5.
- [29] B. Gustavsen and C. Heitz, "Modal vector fitting: A tool for generating rational models of high accuracy with arbitrary terminal conditions," *IEEE Trans. Adv. Pack.*, vol. 31, no. 4, pp. 664–672, Nov. 2008.
- [30] B. Gustavsen, "Passivity enforcement of rational models via modal perturbation," *IEEE Trans. Power Del.*, vol. 23, no. 2, pp. 768–775, Apr. 2008.
- [31] B. Gustavsen, "Rational modeling of multi-port systems via a symmetry and passivity preserving mode-revealing transformation," *IEEE Trans. Power Del.*, vol. 29, no. 1, pp. 199–205, Feb. 2014.

**Bjørn Gustavsen** (Fellow, IEEE) was born in Norway in 1965. He received the M.Sc. and the Dr. Ing. degrees in electrical engineering from the Norwegian Institute of Technology (NTNU), Trondheim, Norway, in 1989 and 1993, respectively. Since 1994, he has been working with SINTEF Energy Research, Trondheim, Norway, where he is currently a Chief Research Scientist. He is also an Adjunct Professor with NTNU, since 2020. He was a Visiting Researcher with the University of Toronto, Toronto, ON, Canada, in 1996, and spent the summer of 1998 with the Manitoba HVDC Research Centre, Winnipeg, MB, Canada. He was a Marie Curie Fellow with the University of Stuttgart, Stuttgart, Germany, August 2001–August 2002. His interests include simulation of electromagnetic transients and modeling of frequency dependent effects.