A Note on the Multiplicity of Poles in the Vector Fitting Macromodeling Method

Dirk Deschrijver and Tom Dhaene, *Senior Member, IEEE*

***Abstract*—Vector fitting is a robust macromodeling tool for rational approximation of spectral data obtained by full-wave electromagnetic simulators or high-frequency measurements. The technique iteratively calculates a suitable set of poles, and solves the residues of the transfer function in a two-step procedure. If poles with a higher order multiplicity occur during the pole-identification step, numerical problems and inaccuracies can result, especially if the normal equations are solved. This problem is illustrated on an *RLC* filter, and a generalization of the basis functions is proposed to resolve the issue.**

***Index Terms*—Macromodeling, rational approximation, vector fitting.**

I. Introduction

VECTOR fitting is a popular macromodeling method, which calculates broadband transfer functions from frequency-domain responses [1], time-domain responses [2], or a combination of both [3]. The robustness of this method is obtained by combining an iterative least squares estimator [4] and some predefined partial fraction basis functions [5]. Initially, these basis functions are based on a prescribed set of poles, which are relocated during successive iterations.

In the original description of the frequency-domain vector fitting method, it is assumed that the prescribed (or relocated) poles always appear with a single multiplicity. For the design of microwave filters, higher order poles are frequently used since they give rise to a sharper peak response at the resonant frequencies [6], [7]. If the poles of the basis functions occur with a higher multiplicity, then the associated system equations become rank deficient and a poor fitting model may result. In this paper, a generalization of the vector fitting method will be described, which solves this issue.

II. Transfer-Function Representation

Frequency-domain macromodeling tools are used to build a rational transfer function $R(s)$, based on the spectral response $(s, H(s))$ of a physical structure

$$R(s) = \frac{N(s)}{D(s)} = \frac{\sum_{p=1}^{P} c_p \varphi_p(s,a)}{\tilde{c}_0 + \sum_{p=1}^{P} \tilde{c}_p \varphi_p(s,a)}. \tag{1}$$

Both the numerator and denominator are expanded as a linear combination of rational basis functions $\varphi_p(s,a)$, which are based on a common set of (prescribed) stable poles $a = \{-a_1, \ldots, -a_P\}$. The goal of the identification process is to identify the coefficients c_p and $\tilde{c}_p$ such that the difference between $R(s)$ and $H(s)$ is minimized in a least squares sense. Levi's linear approximation of this nonlinear identification problem can be obtained by solving the following equations in terms of the unknowns ($\tilde{c}_0 = 1$) in a least squares sense [8]:

$$\sum_{p=1}^{P} c_p \varphi_p(s,a) - H(s)\left(\sum_{p=1}^{P} \tilde{c}_p \varphi_p(s,a)\right) \approx H(s). \tag{2}$$

The basis functions $\varphi_p(s,a)$ of the numerator and denominator are chosen to be partial fractions (vector fitting [1]), or orthonormal rational functions (orthonormal vector fitting [9]). In the vector fitting method, (2) is equivalent to solving

$$\sum_{p=1}^{P} \frac{c_p}{s+a_p} - H(s)\left(\sum_{p=1}^{P} \frac{\tilde{c}_p}{s+a_p}\right) \approx H(s). \tag{3}$$

An implicit version of the Sanathanan–Koerner iteration [4] can then be applied. This means that, in the next iteration, the denominator $D(s)$ is used as an inverse weighting to relieve the bias of Levi's estimator, as shown in (4) at the bottom of the following page. In practice, this means that the prescribed poles a in (3) are replaced by the zeros of $D(s)$, which are the relocated poles of the transfer function. It is known that convergence of this relocation typically occurs in a few iterations, provided that the initial set of poles is well chosen. It is, however, noted that the zeros of $N(s)$ and $D(s)$ may occur with a multiplicity higher than 1, even though the prescribed poles a are all single. This may have some consequences, as will be shown in Section III.

III. Poles With Higher Order Multiplicity

Theorem 1: If a partial fraction with a nonzero residue c_p is based on a pole $-a_p$ with multiplicity $V > 1$, then it cannot be decomposed in a finite set of single-pole fractions.

Proof: A proof of this theorem can be found in several complex analysis books, see, e.g., [10]. ■

From Theorem 1, it follows that the basis function expansion, as described in (3), is not adequate if the poles of the rational basis functions have a higher order multiplicity.

Let M represent the amount of distinct real (M_r) and distinct complex conjugate pairs (M_c) of zeros of denominator $D(s)$ with $M = M_r + 2M_c \leq P$. Also, let $v_{p,d,r}^{(t)}$ $\left(v_{p,d,c}^{(t)}\right)$ represent the multiplicity of the pth real (complex) zero of denominator $D(s)$ at iteration step t of the Sanathanan–Koerner iteration. A generalized expression of the system equations is then derived

Manuscript received October 29, 2006; revised December 12, 2006 and January 10, 2007. This work was supported under the Fund for Scientific Research Flanders (FWO-Vlaanderen).

The authors are with the Department of Mathematics and Computer Science, University of Antwerp, 2020 Antwerp, Belgium (e-mail: dirk.deschrijver@ua.ac.be; tom.dhaene@ua.ac.be).

Digital Object Identifier 10.1109/TMTT.2007.893651

0018-9480/$25.00 © 2007 IEEE

from the least squares formulation (4)–(7), shown at the bottom of this page. Note that the classical vector fitting formulation can easily be derived by setting $v_{p,d,r}^{(t)}$ and $v_{p,d,c}^{(t)}$ equal to unity.

After identification of e_{pm}, $\tilde{e}_{pm}$, f_{pm}, and $\tilde{f}_{pm}$, the poles of the next iteration are calculated by solving the zeros of $D^{(t)}(s)/D^{(t-1)}(s)$ (the so-called "sigma function" in [1]). This term consists of the sum of the following three smaller parts.

- The constant value 1.
- The sum of partial fractions, based on real poles with multiplicity $v_{p,d,r}^{(t-1)}$

$$\sum_{p=1}^{M_r} \sum_{m=1}^{v_{p,d,r}^{(t-1)}} \frac{\tilde{e}_{pm}^{(t)}}{\left(s + z_{p,d,r}^{(t-1)}\right)^m}. \tag{8}$$

- The sum of partial fractions, based on complex conjugate pairs of poles with multiplicity $v_{p,d,c}^{(t-1)}$

$$\sum_{p=1}^{M_c} \sum_{m=1}^{v_{p,d,c}^{(t-1)}} \left(\frac{\tilde{f}_{pm}^{(t)}}{\left(s_k + z_{p,d,c}^{(t-1)}\right)^m} + \frac{\tilde{f}_{pm}^{(t)*}}{\left(s_k + z_{p,d,c}^{(t-1)*}\right)^m} \right) \tag{9}$$

which is equivalent to

$$\sum_{p=1}^{M_c} \left(\sum_{m=1}^{v_{p,d,c}^{(t-1)}} \frac{\tilde{f}_{pm}^{(t)}}{\left(s_k + z_{p,d,c}^{(t-1)}\right)^m} + \sum_{m=1}^{v_{p,d,c}^{(t-1)}} \frac{\tilde{f}_{pm}^{(t)*}}{\left(s_k + z_{p,d,c}^{(t-1)*}\right)^m} \right). \tag{10}$$

These zeros can be calculated by solving an eigenvalue problem, which is based on the real-valued minimal state-space realization of this function (see [1]). The realization of (8) is described in Section IV-A, and the realization of (9) is described in Section IV-B. Based on the state-space representation of these smaller sections, it is possible to construct the realization of the compound expression, as shown in Section IV-C.

IV. State-Space Realization

For ease of representation, the following shorthand notations $V \equiv v_{p,d,r}^{(t-1)}$ and $W \equiv v_{p,d,c}^{(t-1)}$ are introduced, which will be used throughout this section.

A. Real Poles

The minimal continuous-time linear time invariant state-space realization

$$sX(s) = \mathrm{A}_p X(s) + \mathrm{B}_p U(s) \tag{11}$$

$$Y(s) = \mathrm{C'}_p X(s) + \mathrm{D}_p U(s) \tag{12}$$

of a partial fraction with higher order multiplicity V

$$\varphi_p\left(s, z_d^{(t-1)}\right) = \frac{\tilde{e}_{pV}^{(t)}}{\left(s + z_{p,d,r}^{(t-1)}\right)^V} \tag{13}$$

can be calculated by cascading $m = 1, \ldots, V$ times the minimal state-space realization of smaller sections [11]

$$\frac{1}{s + z_{p,d,r}^{(t-1)}} \rightarrow \cdots \rightarrow \frac{1}{s + z_{p,d,r}^{(t-1)}} \rightarrow \frac{\tilde{e}_{pV}^{(t)}}{s + z_{p,d,r}^{(t-1)}}. \tag{14}$$

The minimal state-space realization (A_p^m, B_p^m, C_p^m, D_p^m) of the single-pole partial fractions is given by $(1 \leq m < V)$

$$\mathrm{A}_p^m = -z_{p,d,r}^{(t-1)} \quad \mathrm{B}_p^m = 1 \quad \mathrm{C}_p^m = 1 \quad \mathrm{D}_p^m = 0 \tag{15}$$

$$\arg \min_{c_p^{(t)}, \tilde{c}_p^{(t)}} \left(\sum_{k=0}^{K} \left| \frac{1}{D^{(t-1)}(s_k)} \right|^2 \left| N^{(t)}(s_k) - H(s_k) D^{(t)}(s_k) \right|^2 \right) \tag{4}$$

$$= \arg \min_{z_{p,n}^{(t)}, z_{p,d}^{(t)}} \left(\sum_{k=0}^{K} \left| \frac{\prod_{p=1}^{P}(s_k + a_p)}{\prod_{p=1}^{M} \left(s_k + z_{p,d}^{(t-1)}\right)^{v_{p,d}^{(t-1)}}} \right|^2 \left| \frac{\prod_{p=1}^{M-1} \left(s_k + z_{p,n}^{(t)}\right)^{v_{p,n}^{(t)}}}{\prod_{p=1}^{P}(s_k + a_p)} - H(s_k) \left(\frac{\prod_{p=1}^{M} \left(s_k + z_{p,d}^{(t)}\right)^{v_{p,d}^{(t)}}}{\prod_{p=1}^{P}(s_k + a_p)} \right) \right|^2 \right) \tag{5}$$

$$= \arg \min_{z_{p,n}^{(t)}, z_{p,d}^{(t)}} \left(\sum_{k=0}^{K} \left| \frac{\prod_{p=1}^{M-1} \left(s_k + z_{p,n}^{(t)}\right)^{v_{p,n}^{(t)}}}{\prod_{p=1}^{M} \left(s_k + z_{p,d}^{(t-1)}\right)^{v_{p,d}^{(t-1)}}} - H(s_k) \left(\frac{\prod_{p=1}^{M} \left(s_k + z_{p,d}^{(t)}\right)^{v_{p,d}^{(t)}}}{\prod_{p=1}^{M} \left(s_k + z_{p,d}^{(t-1)}\right)^{v_{p,d}^{(t-1)}}} \right) \right|^2 \right) \tag{6}$$

$$\begin{aligned} = \arg \min_{e_{pm}^{(t)}, \tilde{e}_{pm}^{(t)}, f_{pm}^{(t)}, \tilde{f}_{pm}^{(t)}} & \left(\sum_{k=0}^{K} \left| \left(\sum_{p=1}^{M_r} \sum_{m=1}^{v_{p,d,r}^{(t-1)}} \frac{e_{pm}^{(t)}}{\left(s_k + z_{p,d,r}^{(t-1)}\right)^m} + \sum_{p=1}^{M_c} \sum_{m=1}^{v_{p,d,c}^{(t-1)}} \left(\frac{f_{pm}^{(t)}}{\left(s_k + z_{p,d,c}^{(t-1)}\right)^m} + \frac{f_{pm}^{(t)*}}{\left(s_k + z_{p,d,c}^{(t-1)*}\right)^m} \right) \right) \right. \right. \\ & \left. \left. - H(s_k) \left(1 + \sum_{p=1}^{M_r} \sum_{m=1}^{v_{p,d,r}^{(t-1)}} \frac{\tilde{e}_{pm}^{(t)}}{\left(s_k + z_{p,d,r}^{(t-1)}\right)^m} + \sum_{p=1}^{M_c} \sum_{m=1}^{v_{p,d,c}^{(t-1)}} \left(\frac{\tilde{f}_{pm}^{(t)}}{\left(s_k + z_{p,d,c}^{(t-1)}\right)^m} + \frac{\tilde{f}_{pm}^{(t)*}}{\left(s_k + z_{p,d,c}^{(t-1)*}\right)^m} \right) \right) \right|^2 \right) \end{aligned} \tag{7}$$

and, similarly, $(m = V)$

$$\mathrm{A}_p^V = -z_{p,d,r}^{(t-1)} \quad \mathrm{B}_p^V = 1 \quad \mathrm{C}_p^V = \tilde{e}_{pV}^{(t)} \quad \mathrm{D}_p^V = 0. \tag{16}$$

The minimal state-space realization of the compound system (14) is then obtained as the cascade construction of the smaller state-space models

$$\begin{aligned}
\mathrm{A}_p &= \begin{bmatrix} \mathrm{A}_p^1 & 0 & \dots & 0 \\ \mathrm{B}_p^2\mathrm{C}_p^1 & \mathrm{A}_p^2 & \dots & 0 \\ \mathrm{B}_p^3\mathrm{D}_p^2\mathrm{C}_p^1 & \mathrm{B}_3\mathrm{C}_p^2 & \dots & 0 \\ \mathrm{B}_p^4\mathrm{D}_p^3\mathrm{D}_p^2\mathrm{C}_p^1 & \mathrm{B}_p^4\mathrm{D}_p^3\mathrm{C}_p^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \mathrm{B}_p^V\mathrm{D}_p^{V-1}..\mathrm{D}_p^2\mathrm{C}_p^1 & \mathrm{B}_p^V\mathrm{D}_p^{V-1}..\mathrm{D}_p^3\mathrm{C}_p^2 & \dots & \mathrm{A}_p^V \end{bmatrix} \\
\mathrm{B}_p &= \begin{bmatrix} \mathrm{B}_p^1 \\ \mathrm{B}_p^2\mathrm{D}_p^1 \\ \mathrm{B}_p^3\mathrm{D}_p^2\mathrm{D}_p^1 \\ \mathrm{B}_p^4\mathrm{D}_p^3\mathrm{D}_p^2\mathrm{D}_p^1 \\ \dots \\ \mathrm{B}_p^V\mathrm{D}_p^{V-1}\dots\mathrm{D}_p^1 \end{bmatrix} \\
\mathrm{C}'_p &= \begin{bmatrix} \mathrm{D}_p^V\dots\mathrm{D}_p^2\mathrm{C}_p^1 \\ \mathrm{D}_p^V\dots\mathrm{D}_p^3\mathrm{C}_p^2 \\ \dots \\ \mathrm{C}_p^V \end{bmatrix}^T \\
\mathrm{D}_p &= \mathrm{D}_p^V\dots\mathrm{D}_p^1
\end{aligned} \tag{17}$$

which reduces to the following expressions:

$$\begin{aligned}
\mathrm{A}_p &= \begin{bmatrix} -z_{p,d,r}^{(t-1)} & 0 & \dots & 0 \\ 1 & -z_{p,d,r}^{(t-1)} & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & -z_{p,d,r}^{(t-1)} \end{bmatrix} \\
\mathrm{B}_p &= \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \\
\mathrm{C}'_p &= \begin{bmatrix} 0 \\ \dots \\ 0 \\ \tilde{e}_{pV}^{(t)} \end{bmatrix}^T \\
\mathrm{D}_p &= 0.
\end{aligned} \tag{18}$$

Note that the state matrix A_p and the input vector B_p are built such that the input-to-state transfer functions $(1 \leq m \leq V)$ contain exactly the following basis functions:

$$\frac{X_m(s)}{U_m(s)} = \frac{1}{\left(s + z_{p,d,r}^{(t-1)}\right)^m}. \tag{19}$$

By modifying the output vector C_p as follows:

$$\mathrm{C}_p = \begin{pmatrix} \tilde{e}_{p1}^{(t)} & \tilde{e}_{p2}^{(t)} & \dots & \tilde{e}_{pV}^{(t)} \end{pmatrix} \tag{20}$$

a state-space realization for the following expression is obtained:

$$\sum_{m=1}^{V} \frac{\tilde{e}_{pm}^{(t)}}{\left(s + z_{p,d,r}^{(t-1)}\right)^m}. \tag{21}$$

A state-space realization of (8) can finally be obtained by applying a parallel connection to the state-space realizations for the functions (21) corresponding to each distinct pole. The state-space model can then be build up recursively as follows for $p = 1, \dots, M_r$ (initially $\mathrm{A}_r, \mathrm{B}_r, \mathrm{C}_r = \{\}$, and $\mathrm{D}_r = 0$):

$$\begin{aligned}
\mathrm{A}_r &= \begin{pmatrix} \mathrm{A}_r & 0 \\ 0 & \mathrm{A}_p \end{pmatrix} \\
\mathrm{B}_r &= \begin{pmatrix} \mathrm{B}_r \\ \mathrm{B}_p \end{pmatrix} \\
\mathrm{C}_r &= \begin{pmatrix} \mathrm{C}_r & \mathrm{C}_p \end{pmatrix} \\
\mathrm{D}_r &= \mathrm{D}_r + \mathrm{D}_p.
\end{aligned} \tag{22}$$

B. Complex Conjugate Pairs

If $-z_{p,d,c}^{(t-1)}$ and $-z_{p+1,d,c}^{(t-1)}$ constitute a complex conjugate pair (i.e., $-z_{p+1,d,c}^{(t-1)} = -z_{p,d,c}^{(t-1)*}$), then (7) is solved in terms of $e_{pm}^{(t)}, \tilde{e}_{pm}^{(t)}, \Re e\left(f_{pm}^{(t)}\right), \Im m\left(f_{pm}^{(t)}\right), \Re e\left(\tilde{f}_{pm}^{(t)}\right)$, and $\Im m\left(\tilde{f}_{pm}^{(t)}\right)$, which simplifies the enforcement of real valuedness. Based on these values, a state-space realization of

$$\sum_{m=1}^{W} \frac{\tilde{f}_{pm}^{(t)}}{\left(s_k + z_{p,d,c}^{(t-1)}\right)^m} + \sum_{m=1}^{W} \frac{\tilde{f}_{pm}^{(t)*}}{\left(s_k + z_{p,d,c}^{(t-1)*}\right)^m} \tag{23}$$

is obtained by applying a parallel connection of their two smaller state-space realizations

$$\begin{aligned}
\hat{\mathrm{A}}_{p,p^*} &= \begin{pmatrix} \mathrm{A}_p & 0 \\ 0 & \mathrm{A}_{p+1} \end{pmatrix} \\
\hat{\mathrm{B}}_{p,p^*} &= \begin{pmatrix} \mathrm{B}_p \\ \mathrm{B}_{p+1} \end{pmatrix} \\
\hat{\mathrm{C}}_{p,p^*} &= \begin{pmatrix} \mathrm{C}_p & \mathrm{C}_{p+1} \end{pmatrix} \\
\hat{\mathrm{D}}_{p,p^*} &= \mathrm{D}_p + \mathrm{D}_{p+1}.
\end{aligned} \tag{24}$$

It is noted that these smaller realizations are complex valued since they are constructed as in (18) and (20). In order to obtain a real-valued state-space realization, a similarity transformation is applied on the state-space model

$$\begin{aligned}
\mathrm{A}_{p,p^*} &= T^{-1}\hat{\mathrm{A}}_{p,p^*}T \\
\mathrm{B}_{p,p^*} &= T^{-1}\hat{\mathrm{B}}_{p,p^*} \\
\mathrm{C}_{p,p^*} &= \hat{\mathrm{C}}_{p,p^*}T \\
\mathrm{D}_{p,p^*} &= \hat{\mathrm{D}}_{p,p^*}.
\end{aligned} \tag{25}$$

Provided that I_W represents the unity matrix of dimension $W \times W$, then the transformation matrix T is denoted by

$$T = \frac{1}{2}\begin{pmatrix} I_W & -jI_W \\ I_W & jI_W \end{pmatrix} \quad T^{-1} = \begin{pmatrix} I_W & I_W \\ jI_W & -jI_W \end{pmatrix}. \tag{26}$$

It then results that the transformed matrices will be of the following form:

$$\begin{aligned}
\mathrm{A}_{p,p^*} &= \begin{pmatrix} \Re e(\mathrm{A}_p) & \Im m(\mathrm{A}_p) \\ -\Im m(\mathrm{A}_p) & \Re e(\mathrm{A}_p) \end{pmatrix} \\
\mathrm{B}_{p,p^*} &= \begin{pmatrix} 2 & 0 & \dots & 0 \end{pmatrix}^T \\
\mathrm{C}_{p,p^*} &= \begin{pmatrix} \Re e(\mathrm{C}_p) & \Im m(\mathrm{C}_p) \end{pmatrix} \\
\mathrm{D}_{p,p^*} &= 0.
\end{aligned} \tag{27}$$

A state-space realization of (9) can finally be obtained by applying a parallel connection to the state-space realizations for the functions (23) corresponding to each distinct pair of complex conjugate poles. The state-space model can then be build up recursively as follows for $p = 1, \ldots, M_c$ (initially $\mathrm{A_c}$, $\mathrm{B_c}$, $\mathrm{C_c} = \{\}$, and $\mathrm{D_c} = 0$):

$$\begin{aligned} \mathrm{A_c} &= \begin{pmatrix} \mathrm{A_c} & 0 \\ 0 & \mathrm{A}_{p,p^*} \end{pmatrix} \\ \mathrm{B_c} &= \begin{pmatrix} \mathrm{B_c} \\ \mathrm{B}_{p,p^*} \end{pmatrix} \\ \mathrm{C_c} &= \begin{pmatrix} \mathrm{C_c} & \mathrm{C}_{p,p^*} \end{pmatrix} \\ \mathrm{D_c} &= \mathrm{D_c} + \mathrm{D}_{p,p^*}. \end{aligned} \tag{28}$$

C. Compound Expression

A real-valued state-space realization of the compound expression (i.e., "the sigma function") is then given by

$$\begin{aligned} \mathrm{A} &= \begin{pmatrix} \mathrm{A}_r & 0 \\ 0 & \mathrm{A_c} \end{pmatrix} \\ \mathrm{B} &= \begin{pmatrix} \mathrm{B}_r \\ \mathrm{B_c} \end{pmatrix} \\ \mathrm{C} &= \begin{pmatrix} \mathrm{C}_r & \mathrm{C_c} \end{pmatrix} \\ \mathrm{D} &= 1 + \mathrm{D}_r + \mathrm{D_c} \end{aligned} \tag{29}$$

where A is a $P \times P$ matrix, B is a $P \times 1$ vector, C is a $1 \times P$ vector, and D is a constant. The zeros of sigma, which become the relocated poles of the transfer function, are found by solving the eigenvalues of the matrix A–BC. Calculating the residues of the transfer function can be done in a completely analogous way.

V. Choice of Matrix Solver

Standard application of the vector fitting routine shows that transfer functions with poles of higher order multiplicity are usually quite well approximated by multiple neighboring poles, which have large residues. The accuracy of the result is partially related to the use of an appropriate matrix solver, called the "backslash operator" in MATLAB. For overdetermined systems of equations, this operator performs a QR decomposition with column pivoting, which is fairly robust in handling ill-conditioned matrices.

In some cases, the use of this solver is not practical for the pole identification. If the rational approximation is based on a lot of data samples, then the associated set of system equations becomes large. Especially if the response at a large number of ports is fitted using a common set of poles, a QR decomposition is often undesired because of excessive computation times. Therefore, one often resorts to the normal equations (NEs) for solving the poles, which is computationally more efficient at the expense of some loss in accuracy.

The following example shows that the use of NEs can cause numerical problems if the relocated poles come too close together. Some numerical results illustrate that the extension of basis functions, as described in this paper, improves the numerical conditioning of the system equations, and can lead to better overall results.

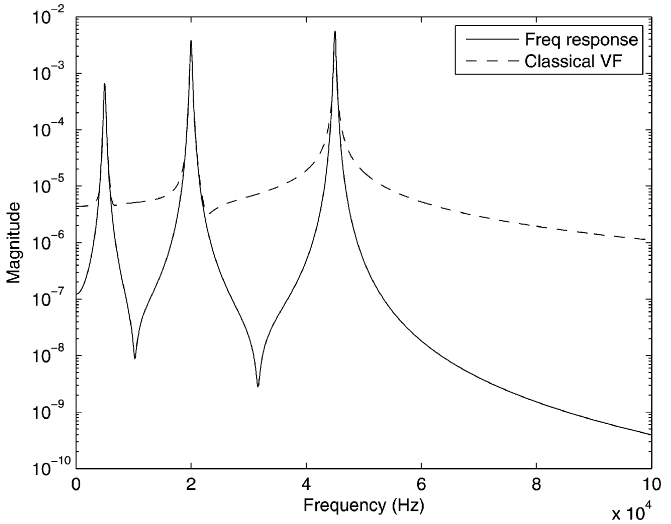

Fig. 1. Magnitude of transfer function (solid line) and classical vector fitting (dashed line). (five iterations, NE/QR). Initial starting poles as described in [1].

VI. Example: *RLC* Filter

In this example, the following frequency response of an *RLC* filter of order 18 is considered over the 1-Hz–100-kHz frequency range:

$$\begin{aligned} H(s) = {} & \frac{40 + j60000}{(s + 220 + j45000)^3} + \frac{40 - j60000}{(s + 220 - j45000)^3} \\ & + \frac{-150 + j40000}{(s + 220 + j20000)^3} + \frac{-150 - j40000}{(s + 220 - j20000)^3} \\ & + \frac{-5 + j7000}{(s + 220 + j5000)^3} + \frac{-5 - j7000}{(s + 220 - j5000)^3}. \end{aligned} \tag{30}$$

The response contains three distinct pairs of complex conjugate poles, each with a higher order multiplicity of 3.

A. Pole Identification

First, an initial set of complex starting poles is chosen with small real parts and with imaginary parts equidistantly spread over the frequency range of interest. It was shown in [1] that this choice gives good results in terms of convergence speed and conditioning. Based on this set of starting poles, the classical vector fitting algorithm is applied. Fig. 1 shows the magnitude of the reference frequency response (solid line), and the resulting fitting model after five iterations (dashed line) if poles are calculated using the NEs, and the residues using a QR decomposition (NE/QR). Clearly, the accuracy of the fitting model is unacceptable, as the convergence process of the fitting technique is significantly impaired. For comparison, the same calculations are performed if poles and residues are both calculated using a QR decomposition (QR/QR). The corresponding rms error in each iteration is shown in Table I.

In successive iterations, it is observed that the poles of the model become relocated towards the correct locations. However, once the correct poles are reasonably well approximated, the system equations of the next pole-identification step become severely ill conditioned. The inaccurate relocation of poles in

TABLE I
rms ERROR DURING SUCCESSIVE ITERATIONS

Iter	Class.VF (NE/QR)	Class.VF (QR/QR)
1	$4.6431 * 10^{-5}$	$2.1501 * 10^{-9}$
2	$2.7491 * 10^{-10}$	$4.5951 * 10^{-12}$
3	$1.7496 * 10^{-6}$	$6.0949 * 10^{-13}$
4	$3.6245 * 10^{-12}$	$1.1727 * 10^{-12}$
5	$6.3425 * 10^{-5}$	$3.3773 * 10^{-13}$

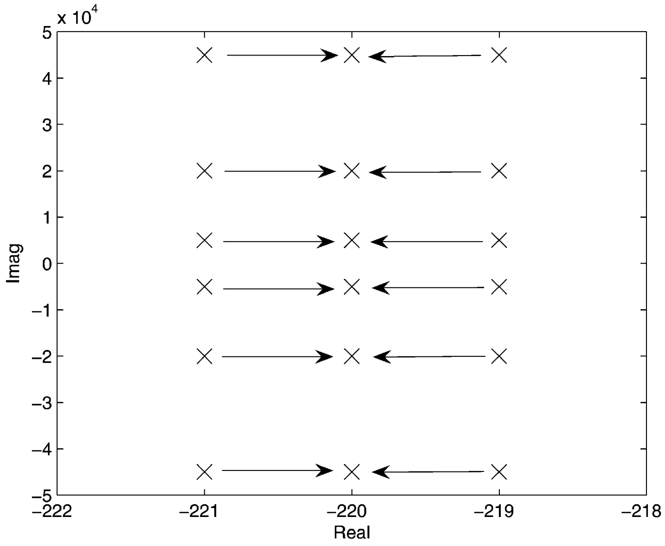

Fig. 2. Pole configuration of basis functions (distance $= 1$).

TABLE II
PROXIMITY OF POLES VERSUS CONDITION NUMBER (POLE IDENTIFICATION)

Method	Distance	Cond. Pole-ident
Classical VF	10	$7.2393 * 10^{+4}$
Classical VF	1	$7.2370 * 10^{+6}$
Classical VF	0.1	$7.2369 * 10^{+8}$
Classical VF	0.01	$7.2369 * 10^{+10}$
Classical VF	0.001	$7.2368 * 10^{+12}$
Classical VF	0.0001	$6.8835 * 10^{+14}$
Classical VF	0	∞
Improved VF	0	6.7708

the following iteration causes a reoccurring relapse in the convergence process. This effect is clearly illustrated by an oscillation error function (NE/QR), shown in Table I. More details about this effect are given in Sections VI-B and C.

B. Pole Perturbation Analysis

Here, the sensitivity of the numerical conditioning is related to the proximity of similar poles. This is analyzed by selecting a set of starting poles for the basis functions, as shown in Fig. 2. Each complex pole (with multiplicity 3) is replaced by three distinct poles, which are slightly perturbed over the real axis. The distance between these poles is systematically reduced in consecutive steps (from 10 to 0), and the corresponding condition number of the pole-relocation system equations is considered in Table II. The condition number grows quickly as the poles are located closer to each other. Knowing that the condition number becomes squared if the NEs are used, one can easily see that rank deficiency occurs, even for poles that are moderately close. The reason for this ill conditioning is that identical (or similar) poles result in identical columns in the system equations. Therefore, it follows from this analysis that the condition number of the pole-identification equations can be significantly improved if clusters of single-multiplicity poles are replaced by a single pole with higher order multiplicity before each iteration. Such clusters can be detected, e.g., by grouping the largest sets of poles for which their mean Euclidean distance in the complex plane is below a predefined threshold. Using this approach, it can be seen from Table III that the oscillating error function is now avoided and more accurate results are obtained. It is important to select this threshold sufficiently small, as additional pole relocations may be required if distant poles are clustered, even though the conditioning of the system equations is improved.

TABLE III
rms ERROR DURING SUCCESSIVE ITERATIONS

Iter	Class.VF (NE/QR)	Impr.VF (NE/QR)
1	$4.6431 * 10^{-5}$	$4.6431 * 10^{-5}$
2	$2.7491 * 10^{-10}$	$2.7491 * 10^{-10}$
3	$1.7496 * 10^{-6}$	$1.2036 * 10^{-10}$
4	$3.6245 * 10^{-12}$	$8.2349 * 10^{-11}$
5	$6.3425 * 10^{-5}$	$2.0249 * 10^{-10}$

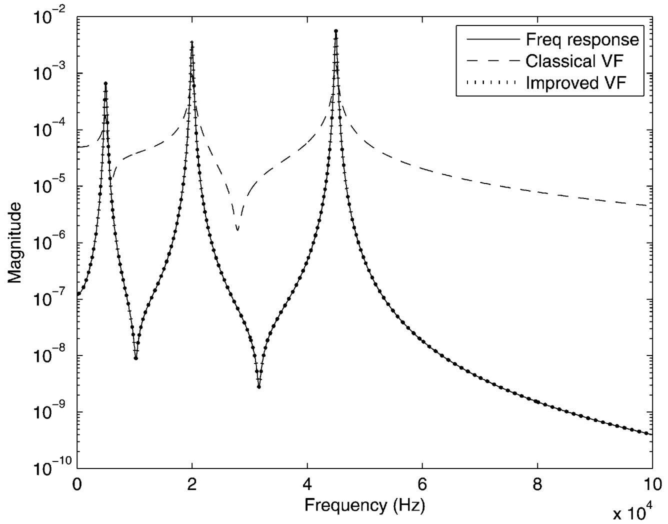

Fig. 3. Magnitude of *RLC* filter (solid line): class. vector fitting (dashed line), improved vector fitting (dotted line). Residue identification using correct poles and QR.

C. Residue Identification

Here, it will be shown that the extended basis functions can also be useful to calculate the residues of the transfer function. Assume that the exact location of the transfer-function poles is known in advance. Based on this prior knowledge, only the residue of each basis function needs to be calculated. Fig. 3 shows that the quality of the fitting model (dashed line) remains unacceptable, even though the poles of the basis functions are exactly known. As shown in Table IV, the inaccuracy is also caused by numerical ill conditioning. If the same calculations are performed using the improved vector fitting technique, based on the extended basis functions as described in this paper, then the model is highly accurate (dotted line). As shown in Table V, the corresponding rms error using the improved vector fitting technique is very close to machine precision. This result

TABLE IV
PROXIMITY OF POLES VERSUS CONDITION NUMBER (RESIDUE IDENTIFICATION)

Method	Distance	Cond. Res-ident
Classical VF	10	$4.4072 * 10^{+3}$
Classical VF	1	$4.4093 * 10^{+5}$
Classical VF	0.1	$4.4093 * 10^{+7}$
Classical VF	0.01	$4.4093 * 10^{+9}$
Classical VF	0.001	$4.4093 * 10^{+11}$
Classical VF	0.0001	$4.4093 * 10^{+13}$
Classical VF	0	∞
Improved VF	0	$1.6118 * 10^{+2}$

TABLE V
rms ERROR OF FITTING MODEL

Method	Normal Equations	QR Decomp
Classical VF	NaN	$3.1608 * 10^{-4}$
Improved VF	$7.4168 * 10^{-18}$	$1.0031 * 10^{-18}$

illustrates that the classical vector fitting technique cannot be applied if some of the basis functions are based on identical poles, even when a QR decomposition is used as matrix solver.

VII. CONCLUSION

It has been shown that the classical vector fitting macromodeling algorithm may deliver a poor fitting model if some poles of the basis functions occur with a higher order multiplicity, especially if the NEs are solved. A generalization of the vector fitting method has been proposed to address this problem, and its effectiveness is illustrated by an example (*RLC* filter).

REFERENCES

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

[2] S. Grivet-Talocia, "Package macromodeling via time-domain vector fitting," *IEEE Microw. Wireless Compon. Lett.*, vol. 13, no. 11, pp. 472–474, Nov. 2003.

[3] S. J. Moon and C. Cangellaris, "Rational function fitting of electromagnetic transfer functions from frequency domain and time domain data," in *IEEE MTT-S Int. Microw. Symp. Dig.*, San Francisco, CA, 2006, pp. 1609–1612.

[4] C. Sanathanan and J. Koerner, "Transfer function synthesis as a ratio of two complex polynomials," *IEEE Trans. Autom. Control*, vol. 8, no. 1, pp. 56–58, Jan. 1963.

[5] W. Hendrickx, D. Deschrijver, and T. Dhaene, "Some remarks on the vector fitting iteration," in *Mathematics in Industry*. Berlin, Germany: Springer-Verlag, 2006, vol. 8, pp. 134–138.

[6] G. L. Matthaei, "Design of wideband (and narrow-band) bandpass microwave filters on the insertion loss basis," *IEEE Trans. Microw. Theory Tech.*, vol. MTT-8, no. 6, pp. 580–593, Jun. 1960.

[7] C. H. Wei, "Design of sharp cutoff low-pass maximally flat *RC*-active filters by cascading third-order blocks," *IEEE Trans. Circuits Syst.*, vol. CAS-27, no. 5, pp. 411–413, May 1980.

[8] E. C. Levi, "Complex curve fitting," *IEEE Trans. Autom. Control*, vol. AC-4, no. 1, pp. 37–43, Jan. 1959.

[9] D. Deschrijver, B. Haegeman, and T. Dhaene, "Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses," *IEEE Trans. Adv. Packaging*, to be published.

[10] R. V. Churchill and J. W. Brown, *Complex Variables and Applications*, 7th ed. New York: McGraw-Hill, 2004.

[11] J. C. Gomez, "Analysis of dynamic system identification using rational orthonormal bases," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Newcastle, Newcastle, Australia, 1998.

Dirk Deschrijver was born in Tielt, Belgium, on September 26, 1981. He received the M.S. degree in computer science from the University of Antwerp, Antwerp, Belgium, in 2003, and is currently working toward the Ph.D. degree at the University of Antwerp.

From May to October 2005, he was a Marie Curie Fellow with the Eindhoven University of Technology, Eindhoven, The Netherlands. He is currently with the Computer Modeling and Simulation (COMS) Group, University of Antwerp. His research interests include rational least squares approximation, orthonormal rational functions, system identification, and macromodeling.

Tom Dhaene (M'02–SM'06) was born in Deinze, Belgium, on June 25, 1966. He received the Ph.D. degree in electrotechnical engineering from the University of Ghent, Gent, Belgium, in 1993.

From 1989 to 1993, he was Research Assistant with the Department of Information Technology, University of Ghent, where his research focused on different aspects of full-wave electromagnetic circuit modeling, transient simulation, and time-domain characterization of high-frequency and high-speed interconnections. In 1993, he joined the Electronic Design Automation (EDA) company Alphabit (now part of Agilent). He was one of the key developers of the planar electromagnetic (EM) simulator ADS Momentum, and he is the principal developer of the multivariate EM-based adaptive metamodeling tool ADS Model Composer. Since September 2000, he has been a Professor with the Department of Mathematics and Computer Science, University of Antwerp, Antwerp, Belgium. He has authored or coauthored over 100 peer-reviewed papers and abstracts in international conference proceedings, journals, and books. He holds two U.S. patents.