Generating High-Accuracy Simulation Models Using Problem-Tailored Orthogonal Polynomials Basis

Carlos Pinto Coelho, *Student Member, IEEE*, Joel R. Phillips, *Senior Member, IEEE*, and Luís Miguel Silveira, *Senior Member, IEEE*

***Abstract*—The problem of computing high-accuracy simulation models for systems described by tabulated frequency data is of paramount importance in the modeling arena. Standard algorithms for this task involve generating rational function approximations to the data. However, for complicated data sets, high-order approximations are required. Unfortunately, numerical conditioning problems arise when attempting to fit high-order rational approximations to the data, effectively limiting the accuracy of the models that can be generated. While robust fitting schemes based on orthogonal polynomial exist, they usually pose strict constraints on the data points, which are either hard or even impossible to guarantee. Furthermore, the approximation must still be translated such that it can directly be used inside a simulator. In this paper, we present an algorithm for robustly generating such a model using only the data given. The model is supported on a problem-tailored orthogonal polynomial basis. We also present a method for directly generating a state-space model associated with a rational function described in terms of such polynomials, effectively making the model amenable for simulation. An extension to the MIMO case is described and it is shown that the method is easily included with existing passivity enforcing procedures. Finally, we demonstrate the proposed technique by constructing approximations to several real-world data sets.**

***Index Terms*—Frequency-data interpolation, orthogonal polynomials basis, problem-specific polynomials, simulation mode.**

I. Introduction

In the design and analysis of communication, high-speed digital, and microwave electronic systems, it is increasingly important to model previously neglected frequency-dependent effects that can have an important impact on the performance and functionality of a design [1]–[3]. For example, in modern communications systems employing advanced digital modulation schemes, such figures of merit as adjacent channel power rejection may depend critically on the frequency-dependent response of the passive circuit components. As a result circuit-level simulation models of the passive components, in particular filters, must give very accurate representations of the actual measurements. However, in the design of microwave and RF communications circuitry, it is often difficult or impractical to directly obtain analytic or numerical models of many passive components, such as surface acoustic-wave filters, spiral inductors, and chip packages, that can be directly incorporated in circuit-level simulators. Instead, the passive devices are characterized by measuring the scattering (S) parameters of physical devices over the frequency range of interest. As another example, in high speed digital systems, impedance (Z), or admittance (Y) parameters describing transmission-line effects [4] are extracted from full-wave electromagnetic field solvers for use in designing clock and power distribution networks. It is sometimes possible to obtain models directly from field solvers [5], but in many cases, engineers are forced to resort to a tedious trial-and-error process of generating, by hand, a lumped circuit model that is sufficiently accurate, yet also compact.

Both examples demonstrate the need for an automatic, accurate, and robust algorithm that can generate simulation models for systems described by sets of tabulated data points. Although it is desirable that such models be as small as possible to simplify their manipulation during the design and analysis phases, more often than not, accuracy considerations make the use of large models unavoidable if one is to achieve a good frequency fit in the model. Unfortunately, high-order models are hard to obtain, as numerical problems arise during their computation. Standard techniques for fitting a model to tabulated frequency data are usually based on determining the coefficients of a rational function approximation. This is typically done with a least squares procedure using the standard monomial basis, i.e., the polynomials $1, s, s^2, \cdots$, etc. For high orders, however, such techniques lead to ill-conditioned problems and are in general not easy or even impossible to apply. Frequency scaling can alleviate this problem to some extent, but for high-order approximation it is known not to be sufficient remedy [6]. Alternative procedures, using rational function approximations (for a very successful technique in this regard, see [7]), suffer from the same overall difficulty, and fail to provide a practical alternative for high-order, high-accuracy approximations. Robust schemes, based on orthogonal polynomial representations are seen as a possible solution to this problem as they can enable robust ways to compute high-order approximants. Much research has been devoted to system identification, namely in the control and systems arena, and several methods using orthogonal polynomials have been proposed. A procedure for performing frequency domain identification using generalized orthonormal basis functions was presented in [8]. However this technique assumes that a time-domain generating system is available, from which frequency data is obtained via transformation. Another class of techniques along the same lines makes use of Chebyshev polynomials, to mitigate the numerical problem [6], [9]. However,

Manuscript received July 21, 2005; revised November 18, 2005. This paper was recommended by Associate Editor C.-T. Lin.

C. P. Coelho is with the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: cfspc@mit.edu).

J. R. Phillips is with Cadence Berkeley Laboratories, Cadence Design Systems Inc., Berkeley, CA 94704 USA (e-mail: jrp@cadence.com).

L. M. Silveira is with Instituto de Engenharia de Sistemas e Computadores, Investigaçáo e Desenvolvimento em Lisboa (INESC-ID), Department of Electrical and Computer Engineering, Instituto Superior Técnico, Technical University of Lisbon, Lisbon 1000–029, Portugal, and also with Cadence Laboratories, Lisbon 1000–029, Portugal, (e-mail: lms@inesc-id.pt).

Digital Object Identifier 10.1109/TCSI.2006.883865

1057-7122/$20.00 © 2006 IEEE

their robustness properties are usually associated with strict constraints on the positioning of the data points, which are either hard or even impossible to guarantee. New data points can, in principle, be generated via interpolation to satisfy such requirements, but this process is complicated and can lead to additional errors. Another potential solution to obviate the numerical issues when approximating data over large frequency ranges, is to partition the frequency axis into smaller ranges and compute approximants in each partition. Several procedures have been proposed for this task [10], [11]. Basically, these algorithms propose constructing an accurate global approximant by piecing together the local partition-generated models, perhaps further enforcing other properties. However, as models computed in each partition also contribute to the frequency response outside that partition, this interaction must be accounted for when constructing the global approximant. This generally requires a more complicated algorithm to control the patching of the various models. Furthermore these algorithms are not entirely useful when handling complicated data in a short frequency range, as the partitioning does not yield a considerable advantage.

In this paper, we present a method for generating an orthogonal polynomial basis tailored to a specific data set. Once the basis is generated it can then be used with most fitting algorithms to robustly generate a highly accurate, high-order model. While the robustness of the procedure is essential to the modeling phase, two additional concerns have to be taken into account regarding the generated model. First, the model must be amenable to simulation in a standard environment which typically requires a translation to a state-space form or equivalent. While the usage of problem-tailored polynomial basis enables the generation of very accurate high-order models, it is essential that such models be represented in a standard form for simulation purposes. Using standard realization procedures is unacceptable since the numerical difficulties mentioned would reappear in this process. Still, we will show that it is possible to generate efficient state-space representations of the model in a robust manner, thus retaining the desired accuracy. A second concern is the related to model physical properties. In order to guarantee good simulation behavior for the generated model, not only must it match the available data but it must also possess stability and passivity properties similar to those of the physical system that it represents. Otherwise, nonphysical anomalies may be introduced into the time-domain simulation leading to erroneous results. Fortunately, this problem has recently been the target of considerable research and methods are now available to generate stable and passive approximations to frequency data [12]–[14]. However, all such methods require as a starting point, a highly accurate initial approximation. In this context, the proposed method can be seen as an efficient enabler or an essential first step for such methods.

We start in Section II with a brief description of the problem of computing rational approximations to tabulated data and we identify the source of numerical ill-conditioning when using higher order models. Then, in Section III, we present an algorithm that, given a set of frequency points, generates an orthogonal polynomial basis on that set of points. We also discuss how to directly generate an efficient state-space representation using the orthogonal polynomial basis description directly, and also how to dramatically improve the choice of a good starting point for the fitting parameters. In Section IV, we discuss the extension of the proposed method to the case of multiple-port systems. In Section V, we present some results where the use of the problem specific orthogonal polynomial basis allows the robust generation of very high-order models. Finally, in Section VI, we draw some conclusions.

II. Rational Approximation

Given a set of data points describing the behavior of a system in the frequency domain, i.e., a sampled transfer function, $\mathbf{H}(s) = \{H(s_i)\}$, $i = 1, \ldots, N$, the rational approximation problem amounts to computing a state-space model

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} + \mathbf{D}u \end{aligned} \tag{1}$$

or its equivalent rational function form

$$\mathbf{R}(\mathbf{a}, \mathbf{b}, s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} = \frac{\sum_{k=0}^{m} b_k s^k}{\sum_{k=0}^{n} a_k s^k} = \frac{\mathbf{Y}(s)}{\mathbf{U}(s)} \tag{2}$$

where $a_0 = 1$, $\mathbf{a} = [1, a_1, \ldots, a_n]^T$, $\mathbf{b} = [b_0, \ldots, b_m]^T$, and such that

$$\mathbf{E}(\mathbf{a}, \mathbf{b}, s) = \mathbf{R}(\mathbf{a}, \mathbf{b}, s) - \mathbf{H}(s) \tag{3}$$

evaluated on the discrete frequency points, is minimized in some appropriate norm.

This nonlinear minimization can for instance be accomplished with a Levenberg–Marquardt procedure, which requires evaluation of the Jacobian of the objective function (see [15] for details). To understand how the use of the monomial basis leads to numerical ill-conditioning consider the evaluation of (2) at the data points or the Jacobian of the objective function, which is often used to minimize (3). Either case involves the product of the parameter vectors $\mathbf{a}$ and $\mathbf{b}$ with the Vandermonde matrix $\mathbf{U}_{1:n}$ or $\mathbf{U}_{0:m}$ respectively. The Vandermonde matrix $\mathbf{U}_{0:m}$ is defined as

$$\mathbf{U}_{0:m} = \begin{bmatrix} 1 & s_1 & s_1^2 & \cdots & s_1^m \\ 1 & s_2 & s_2^2 & \cdots & s_2^m \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & s_N & s_N^2 & \cdots & s_N^m \end{bmatrix} = \left[\mathbf{1}, \mathbf{S}\mathbf{1}, \mathbf{S}^2\mathbf{1}, \ldots, \mathbf{S}^m\mathbf{1}\right] \tag{4}$$

where $\mathbf{S} = \mathrm{diag}([s_1, s_2, \ldots, s_N]^T)$, is a diagonal matrix whose entries are the various frequency points, and $\mathbf{1}$ is an N row column vector with all entries set to 1. The Vandermonde matrix is known to lose numerical rank very fast [16]. Fig. 1 illustrates this problem. Here, we plot the condition number of the Vandermonde matrix as a function of the number of data points in two situations. This loss of rank means that, for relatively low values of m, there will be $\mathbf{b} \neq 0$ such that $\mathbf{U}_{0:m}\mathbf{b} = \mathbf{0}$. The same holds for n, $\mathbf{U}_{1:n}$ and $\mathbf{a}$. This means that even though we increase the model order we will get no better fit; it also means that the function and Jacobian evaluation will be riddled with numerical error as the basis functions will not be accurately represented.

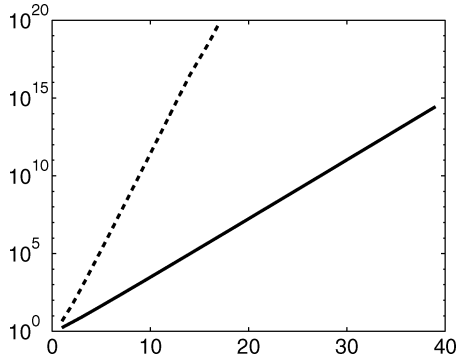

Fig. 1. Condition number of optimally scaled matrix $\mathbf{U}_{0:m}$ as a function of m. Results for linearly (solid line) and logarithmically (dashed line) spaced points are shown.

As previously mentioned, it is possible to fit the data points to an alternative representation that avoids the numerical difficulties discussed. A possible candidate would be to replace the monomial polynomial basis used in (2) with Chebyshev polynomials. This would lead to a representation such as

$$\hat{\mathbf{R}}(\hat{\mathbf{a}}, \hat{\mathbf{b}}, s) = \frac{\sum_{k=0}^{m} \hat{b}_k T_k(s)}{\sum_{k=0}^{n} \hat{a}_k T_k(s)} = \frac{\mathbf{Y}(s)}{\mathbf{U}(s)} \tag{5}$$

where $T_k(s)$ is the kth-order Chebyshev polynomial of the first kind, satisfying a three-term relation such as

$$T_{k+1}(s) = 2sT_k(s) - T_{k-1}(s)$$

and $\hat{a}_0 = 1$, $\hat{\mathbf{a}} = [1, \hat{a}_1, \ldots, \hat{a}_n]^T$, $\hat{\mathbf{b}} = [\hat{b}_0, \ldots, \hat{b}_m]^T$ are coefficients of the approximations, which can be computed from a minimization procedure such as the one described. Chebyshev polynomials possess the important property that they satisfy a discrete orthogonality condition at the polynomials zeros and lead to better conditioned properties, *when the data set also conforms to these zero locations*. It is important to realize that the basis obtained from evaluating Chebyshev, or other orthogonal polynomials, at arbitrary points is *not* well conditioned. Thus, in order to take advantage of such properties, restrictions on the data points used in the fitting procedure must be imposed, which is either impossible to do or may lead to additional error generation.

III. Using Alternative Polynomial Basis

A. Robust Basis Generation

To deal with the numerical rank deficiency problem of the Vandermonde matrix without constraining the placement of the datapoints, we start by noticing that the columns of the matrix span a Krylov subspace with generating matrix $\mathbf{S}$ and starting vector $\mathbf{1}$, $\mathbf{K}_{k+1}(\mathbf{S}, \mathbf{1})$, as is clear from (4). It is well known that such a basis leads to ill-conditioned representations of the subspace. Therefore, we propose instead to generate an orthonormal polynomial basis that spans the same space as the columns of the ideal Vandermonde matrix. An orthonormal basis for $colspan\{\mathbf{U}_{0:k}\}$ can readily be generated using an Arnoldi process—the Arnoldi process produces the factorization

$$\mathbf{S}\mathbf{V}_k = \mathbf{V}_k \mathbf{H}_k + \mathbf{v}_{k+1} h_{k+1,k} \tag{6}$$

where $\mathbf{V}_k$ has orthonormal columns (i.e., $\mathbf{V}_k^H \mathbf{V}_k = \mathbf{I}_k$, $\mathbf{V}_k^H$ stands for conjugate transpose of $\mathbf{V}_k$ and $\mathbf{I}_k$ is the identity matrix of size k), $\mathbf{H}_k$ is a $k \times k$ upper Hessenberg matrix (see [17], [18] for more details) and $\mathbf{V}_{0:k} = [\mathbf{V}_k, \mathbf{v}_{k+1}]$ is the orthonormal basis we seek.

Notice that since $\mathbf{S}$ is anti-Hermitian so is $\mathbf{H}_k$. But $\mathbf{H}_k$ is also upper Hessenberg by definition. Therefore, it has to be a tridiagonal matrix. In this case the Arnoldi process becomes a Lanczos process and the cost of generating an order k orthogonal basis is $\mathcal{O}(Nk)$.

In other words, this is a process for constructing a polynomial basis defined by the three-term recursion

$$sP_{j-1}(s) = \sum_{i=1}^{j+1} h_{i,j} P_{i-1}(s) = \sum_{i=j-1}^{j+1} h_{i,j} P_{i-1}(s) \tag{7}$$

where $P_j(s)$ is a polynomial of order j. The polynomial basis thus defined satisfies the discrete orthogonality relation

$$\sum_{k=1}^{N} P_j(s_k) P_i(s_k) = \delta_{ij} \tag{8}$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if and only if $i = j$, 0 otherwise). If necessary, the above relation can be replaced by a weighted discrete orthogonality condition. The orthogonality is then enforced using basis polynomials that, after weighting, are orthogonal on the frequency points. This can be accomplished by starting the recursion with a weight vector $W^{1/2}$ and then dividing the elements in each column of $\mathbf{V}_k$ by $W^{1/2}$. Such a procedure would still guarantee orthogonality, essential for numerical reasons, and would also allow some flexibility in terms of enforcing greater accuracy in certain frequency ranges. The usual technique to accomplish this would be to introduce some frequency weighting directly into (3), which is undesirable in this case as it would destroy the orthogonality. Using the above procedure we are effectively weighting the minimization but the matrices involved will be orthogonal by construction.

In order to guarantee that the model has a real time-domain response, its frequency response must obey the complex conjugate symmetry condition. This requires that the polynomial basis we are generating also satisfy this condition and that $\mathbf{a}$ and $\mathbf{b}$ be real vectors. This can be achieved by guaranteeing that the basis polynomials are orthogonal on both sides of the frequency axis. This corresponds to generating $\mathbf{K}_{k+1}(\mathbf{S}, \mathbf{1})$ where now $\mathbf{S} = \mathrm{diag}([s; \overline{s}])$, $\overline{s}$ is a complex conjugate of s, and $\mathbf{1}$ would be a $2N$ column vector with all entries set to 1. In practice, a similar result can be obtained by using a modified Arnoldi process where only the real projections are used, thus generating a real coefficient polynomial basis which is orthogonal when both sides of the axis are considered. In the following, in order not to make the notation confusing, we assume that one of these options is implicit.

Replacing the Vandermonde matrices by the new basis matrices means that function and Jacobian evaluations will depend on $\mathbf{V}_{1:n}\mathbf{a}$ and $\mathbf{V}_{0:m}\mathbf{b}$ which greatly increases the numerical stability of the function evaluation and approximation procedure.

The increased numerical stability of these processes allows the creation of very high-order models, a task that is impossible in double precision using the monomial basis and the Vandermonde matrices. By changing from the monomial base to the new polynomial basis, (2) becomes

$$\tilde{\mathbf{R}}(\tilde{\mathbf{a}},\tilde{\mathbf{b}},s)=\frac{\sum_{k=0}^{m}\tilde{b}_k P_k(s)}{\sum_{k=0}^{n}\tilde{a}_k P_k(s)}=\frac{\mathbf{Y}(s)}{\mathbf{U}(s)} \tag{9}$$

where the $P_j(s)$ are given by (7).

Equation (9) represents the, potentially high-order, accurate model of our system. However, as mentioned previously, the goal is not to generate the model but to use it in the context of some global system analysis or verification. As such, it is necessary to find an alternative representation that is amenable to simulation in the time domain. A solution to this problem would be to convert the model in (9) to some standard form from which a standard state-space representation can be derived. Such a representation can be included in a simulator, either directly or by synthesizing an equivalent circuit. However, such an approach either requires converting back to the monomial basis, which we obviously want to avoid, or convert to a pole–zero representation, an expensive procedure requiring the solution of an eigenvalue problem. Neither of the options is very appealing. To circumvent this problem it is therefore essential that in the process of generating a state-space representation, our representation is kept as a function of the alternative basis. In the following section, we reconcile these two requirements.

B. Generating State-Space Models

In order to obtain a state-space representation that can be used to simulate the system in the time domain, we developed a generalized controller canonical form for polynomials defined by recursion relations. To simplify, consider $m<n$; writing (9) as

$$\mathbf{Y}(s)=\left(\sum_{k=0}^{m}\tilde{b}_k P_k(s)\right)\left(\sum_{k=0}^{n}\tilde{a}_k P_k(s)\right)^{-1}\mathbf{U}(s) \tag{10}$$

and defining $\boldsymbol{\xi}(s)=(\sum_{k=0}^{n}\tilde{a}_k P_k(s))^{-1}\mathbf{U}(s)$ and $F_j(s)=P_j(s)\boldsymbol{\xi}(s)$ leads to

$$\begin{aligned}\sum_{k=0}^{m}\tilde{b}_k\left(P_k(s)\boldsymbol{\xi}(s)\right)&=\sum_{k=0}^{m}\tilde{b}_k F_k(s)=\mathbf{Y}(s)\\ \sum_{k=0}^{n}\tilde{a}_k\left(P_k(s)\boldsymbol{\xi}(s)\right)&=\sum_{k=0}^{n}\tilde{a}_k F_k(s)=\mathbf{U}(s).\end{aligned} \tag{11}$$

Applying the inverse Laplace transform to (11) then leads to

$$\begin{aligned}\sum_{k=0}^{m}\tilde{b}_k f_k(t)&=y(t)\\ \sum_{k=0}^{n}\tilde{a}_k f_k(t)&=u(t)\end{aligned} \tag{12}$$

where $f_k(t)$ is the inverse transform of $F_k(s)$.

Finally, multiplying (7) by $\boldsymbol{\xi}(s)$ and applying the inverse Laplace transform, we get

$$\frac{df_{j-1}(t)}{dt}=\sum_{i=j-1}^{j+1}h_{i,j}f_{i-1}(t) \tag{13}$$

which in our particular case, recalling that $\mathbf{H}$ is anti-Hermitian, leads to

$$\frac{df_{j-1}(t)}{dt}=h_{j+1,j}f_j(t)+h_{j-1,j}f_{j-2}(t),$$

i.e. a three-term recurrence similar the one used to define Chebyshev polynomials [19]. From (12), we know that there is a linear relationship between the various $f_k(t)$. This allows us to stop the growing recursion by expressing $f_n(t)$, for instance, as a linear function of the other $f_k(t)$, $k<n$ and the input, $u(t)$,

$$f_n(t)=-\frac{1}{\tilde{a}_n}\sum_{k=0}^{n-1}\tilde{a}_k f_k(t)+\frac{1}{\tilde{a}_n}u(t). \tag{14}$$

We can then use the $f_j(t)$ as state-space variables to construct the state-space model. The state-space equations lead to

$$\left\{\begin{array}{rcl}\frac{d}{dt}f_0(t) &=& h_{0,1}f_1(t)\\ \frac{d}{dt}f_1(t) &=& h_{0,1}f_0(t)+h_{2,1}f_2(t)\\ &\vdots&\\ \frac{d}{dt}f_{n-2}(t) &=& h_{n-3,n-2}f_{n-3}(t)+h_{n-1,n-2}f_{n-1}(t)\\ \underbrace{\frac{d}{dt}f_{n-1}(t)} &=& h_{n-2,n-1}f_{n-2}(t)\\ \multicolumn{3}{c}{\scriptstyle \frac{d}{dt}\mathbf{f}_{(t)}=\mathbf{H}^T\mathbf{f}}\\ \multicolumn{3}{l}{+h_{n,n-1}f_n(t)}\end{array}\right. \tag{15}$$

where $\mathbf{f}=[f_0,f_1\cdots f_{n-1}]^T$. Equation (15) can now be used to express $f_n(t)$ in (14) in terms of the state-space unknowns. Recalling the standard state-space description from (1), and using the frst equation in (12) as the output equation, this can be written compactly in matrix terms as

$$\begin{aligned}\mathbf{A}&=\mathbf{H}_n^T-\tilde{a}_n^{-1}h_{n+1,n}\mathbf{e}_n[\tilde{a}_0\cdots\tilde{a}_{n-1}]\\ \mathbf{B}&=\tilde{a}_n^{-1}h_{n+1,n}\mathbf{e}_n\\ \mathbf{C}&=[\tilde{b}_0\cdots\tilde{b}_m 0\cdots 0]\\ D&=0\end{aligned} \tag{16}$$

which can then be used as the system model for time-domain simulations.

C. Accelerating Nonlinear Optimization

The previous derivation avoids the loss of rank associated with the standard monomial representation and enables the generation of very high-order, highly accurate models. It essentially deals with the issue of the inner loop of the minimization of (3). Still, such a minimization is very sensitive to the choice of an adequate starting point for the iteration. We dealt with this problem by using an heuristic method which we found to converge quickly to a good starting point but which demonstrated poor local convergence properties.

An alternative to the minimization of (3), which leads to a nonlinear minimization, is to cast the problem as a linear minimization. For instance, in the simplified case of approximating a scalar transform function, to linearize the nonlinear rational function fitting problem we multiply both sides of (3) by the denominator of the rational approximation, i.e.

$$\min \|\mathbf{E}(a,b,s)\|_2 = \min \left\| \sum_{k=0}^{m} b_k s^k - H \sum_{k=0}^{n} a_k s^k \right\|. \quad (17)$$

Minimization of (17) is now equivalent to a standard linear least-squares problem whose solution is well known.

The same problem can now be written, using our new orthogonal basis, as

$$\min \tilde{\mathbf{E}}(\tilde{a},\tilde{b},s) = \min \left\| \sum_{k=0}^{m} \tilde{b}_k P_k(s) - \mathbf{H} \sum_{k=0}^{n} \tilde{a}_k P_k(s) \right\| \quad (18)$$

This problem can be put in matrix form and the results described in the previous section can be directly applied to it. Doing this leads to

$$\begin{bmatrix} Re\{\mathbf{V}_{0:m}\} \vdots Re\{-\mathbf{H}\mathbf{V}_{1:n}\} \\ Im\{\mathbf{V}_{0:m}\} \vdots Im\{-\mathbf{H}\mathbf{V}_{1:n}\} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{b}} \\ \tilde{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} Re\{\mathbf{H}\mathbf{V}_{1:n}\} \\ Im\{\mathbf{H}\mathbf{V}_{1:n}\} \end{bmatrix} \quad (19)$$

where $[\tilde{\mathbf{b}}\ \tilde{\mathbf{a}}]^T = [\tilde{b}_0 \cdots \tilde{b}_m\ \tilde{a}_1 \cdots \tilde{a}_n]^T$.

The solution of (19) can then be used as a starting point for the original nonlinear minimization, or incorporated in any sort of fast hybrid scheme (see [15] for a suggestion and details).

A very simplified version of the proposed interpolation algorithm is shown as Algorithm 1.

Algorithm 1: **Interpolation Using Problem-Tailored Orthogonal Basis Algorithm**

1) Generate an orthogonal polynomial basis for the data set using (6).
2) Determine the interpolant error in an appropriate norm

$$\min_{\tilde{\mathbf{a}},\tilde{\mathbf{b}}} \left\| \tilde{\mathbf{R}}(\tilde{\mathbf{a}},\tilde{\mathbf{b}},s) - \mathbf{H}(s) \right\|$$

at the sample points s_i, $i = 1,\ldots,N$.
   a) Generate an initial approximation for the nonlinear minimization: solve (18) (not necessarily to convergence).
   b) Improve the initial approximation using a nonlinear optimization algorithm such as Levenberg–Marquardt (to convergence).
3) Generate a state-space presentation of the model system using (16).

IV. Multiport Models

Because it is necessary to model loading effects, most systems encountered in RF design are multi-input multi-output (MIMO). We now consider generalizing the above algorithms to MIMO data for a q-input, p-output system, $\mathbf{Y}(s) \in \mathbb{C}^{p\times q}$. Generally speaking, we have encountered a tradeoff between compactness of the MIMO representation and robustness of the approximation process.

Perhaps the easiest way to generate a MIMO model is to approximate each of the pq entries of the data $\mathbf{Y}(s)$ matrices as single, separate and independent single-input single-output (SISO) systems. To accomplish this, one would pick up all the data points corresponding to a single (i,j) matrix entry and would generate a model for that string of data. Overall, this leads to pq models, each of which approximates a different frequency-dependent entry of the matrix transfer function. All of the algorithms of the previous section can be easily applied without modification. Once all models are generated, they could be collected to form a single state-space model. To see how such a model can be realized from the separate SISO systems consider the 2×2 case. Suppose state-space model ABCD matrices have been generated for each of the matrix transfer function entries, with $[\mathbf{A}_{11},\mathbf{B}_{11},\mathbf{C}_{11},\mathbf{D}_{11}]$ the model for the $\mathbf{Y}_{11}(s)$, and similarly for $\mathbf{Y}_{12}(s)$, etc.

The matrix transfer function can be realized by $[\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}]$ if

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & & & \\ & \mathbf{A}_{21} & & \\ & & \mathbf{A}_{12} & \\ & & & \mathbf{A}_{22} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & 0 \\ \mathbf{B}_{21} & 0 \\ 0 & \mathbf{B}_{12} \\ 0 & \mathbf{B}_{22} \end{bmatrix} \quad (20)$$

$$\mathbf{C}^T = \begin{bmatrix} \mathbf{C}_{11}^T & 0 & \mathbf{C}_{12}^T & 0 \\ 0 & \mathbf{C}_{21}^T & 0 & \mathbf{C}_{22}^T \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix}. \quad (21)$$

Unfortunately, such a realization contains a potentially large number of states and may include considerable redundancy. If each transfer function requires n states to approximate, the total number of states is npq which grows quadratically with the number of input/output pairs. Furthermore, many of the individual transfer functions may share the same dynamics, leading to a very redundant representation. The resulting model may therefore be far from an efficient realization, and a follow-on model reduction step may be necessary to obtain a model compact enough to use in time-domain simulation. However, for high-order models (e.g., $m = 100$) with many ports (more than two or three inputs and outputs) the model reduction can represent the dominant part of the computational cost, and in fact may be impractical, particularly if a computationally demanding optimal technique is used [20]. In addition, for models with unusual scaling, it can be difficult to find a model reduction approach with both adequate accuracy and compression. Therefore, it is not wise to simply discard these issues during model construction assuming that later on some reduction algorithm will be able to compress the generated model. Instead it is worthwhile to seek rational approximation algorithms that can generate models of reasonable dimensions, even if later compression is still attempted.

At the opposite extreme of the multiple SISO approach just described, are MIMO models written with matrix rational function notation [21]. We will first express the relationship between the coefficients of a matrix rational function

$$\mathbf{Y}(s) = \left(\sum_{k=0}^{m} \mathbf{B}_k P_k(s)\right)\left(\sum_{k=0}^{n} \mathbf{A}_k P_k(s)\right)^{-1} \mathbf{U}(s) \quad (22)$$

$\mathbf{Y}(s) \in \mathbb{C}^q$ with $\mathbf{U}(s) \in \mathbb{C}^q$, $\mathbf{A}_k \in \mathbb{R}^{q\times q}$ and $\mathbf{B}_k \in \mathbb{R}^{p\times p}$, and the corresponding state-space model. Defining $\boldsymbol{\xi}(s) = (\sum_{k=0}^{n} \mathbf{A}_k P_k(s))^{-1}\mathbf{U}(s) \in \mathbb{C}^p$ and $\mathbf{F}_j(s) = P_j(s)\boldsymbol{\xi}(s) \in \mathbb{C}^p$ we get

$$\mathbf{Y}(s) = \sum_{k=0}^{m} \mathbf{B}_k \mathbf{F}_k(s)$$
$$\sum_{k=0}^{n} \mathbf{A}_k \mathbf{F}_k(s) = \mathbf{U}(s). \quad (23)$$

As in the SISO case, we can use these equations to close the polynomial recursion and generate a state-space model. In this case, this leads to

$$\begin{aligned}
\mathbf{A} &= \mathbf{H}_n^T \otimes \mathbf{I}_j - h_{n+1,n}(\mathbf{e}_n \otimes \mathbf{I}_j)\left[\mathbf{A}_n^{-1}\mathbf{A}_0, \ldots, \mathbf{A}_n^{-1}\mathbf{A}_{n-1}\right] \\
\mathbf{B} &= h_{n+1,n}(\mathbf{e}_n \otimes \mathbf{I}_j)\mathbf{A}_n^{-1} \\
\mathbf{C} &= [\mathbf{B}_0, \ldots, \mathbf{B}_m 0, \ldots, 0] \\
\mathbf{D} &= 0
\end{aligned} \quad (24)$$

where $\otimes$ is the Kronecker product [22]. We point out that (16) and (24) are analogous and very similar.

The linearized system also generalizes to the MIMO case

$$\mathbf{H}(s) = \left(\sum_{k=0}^{m} \mathbf{B}_k P_k(s)\right)\left(\sum_{k=0}^{n} \mathbf{A}_k P_k(s)\right)^{-1}. \quad (25)$$

Fixing $\mathbf{A}_0 = \mathbf{I}_p$ we get

$$\mathbf{H}(s)P_0(s) = \sum_{k=0}^{m} \mathbf{B}_k P_k(s) - \mathbf{H}(s)\sum_{k=1}^{n} \mathbf{A}_k P_k(s) \quad (26)$$

to solve for $\mathbf{A}_k$ and $\mathbf{B}_k$ in the least squares sense. Again, this can be represented as an overdetermined system of linear equations to be split in real and imaginary parts and solved in the least squares sense.

This representation is effective for generating relatively small models of data whose matrix entries have similar scaling, or when relatively good estimates for the matrix denominator can be obtained easily. Problems occur, however, on MIMO data where the matrix entries have very different scaling. For example, consider the two-port S-parameters of a low-loss, high-rejection bandpass filter. In the passband, the transmission $|s_{12}|$ is large, and the reflected power s_{11} small. In the stop band, the converse is true. The ratio $|s_{12}/s_{11}|$ can thus span many decades which may lead to conditioning problems if the matrix entries have very different scaling.

A compromise between the full MIMO model representation, which is compact but difficult to converge to high precision, and the shared-denominator form of (21), which is easy to converge but not compact, is a representation of a matrix transfer function "by columns." In this representation each column of a matrix transfer function

$$\mathbf{H}(s) = \begin{bmatrix} \mathbf{H}_{11}(s) & \mathbf{H}_{12}(s) & \ldots & \mathbf{H}_{1k}(s) \\ \mathbf{H}_{21}(s) & \ddots & & \\ \vdots & & & \\ \mathbf{H}_{k1}(s) & \mathbf{H}_{k2}(s) & \ldots & \mathbf{H}_{kk}(s) \end{bmatrix} \quad (27)$$

is represented as a separate state-space model, e.g.

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{H}_{11}(s) \\ \mathbf{H}_{21}(s) \\ \ddots \\ \mathbf{H}_{k1}(s) \end{bmatrix} = \mathbf{C}_1(s\mathbf{I} - \mathbf{A}_1)^{-1}\mathbf{B}_1 + \begin{bmatrix} \mathbf{D}_{11} \\ \mathbf{D}_{21} \\ \ddots \\ \mathbf{D}_{k1} \end{bmatrix}. \quad (28)$$

The columns can be assembled into a single state-space model

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})\mathbf{B} + \mathbf{D} \quad (29)$$

if $\mathbf{A}, \mathbf{B}, \mathbf{C}$ (there being no latitude in construction of $\mathbf{D}$) are constructed as

$$\begin{aligned}
\mathbf{C} &= \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \cdots \\ \mathbf{C}_k \end{bmatrix}^T \\
\mathbf{A} &= \begin{bmatrix} \mathbf{A}_1 & 0 & \cdots & \\ 0 & \mathbf{A}_2 & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & & & \mathbf{A}_k \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} \mathbf{B}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{B}_2 & \cdots & 0 \\ 0 & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{B}_k \end{bmatrix}.
\end{aligned} \quad (30)$$

This procedure is quite efficient and also shows that it is possible to compute a state-space representation of our frequency-described MIMO system, working exclusively in the proposed robust polynomial basis. The vector representation allows us to use hybrid fitting algorithms which have been shown to have very fast global convergence time and yet achieve a reasonably compact final model (see [15] for presentation and discussion). This is an important tradeoff in practical applications.

V. Results

In this section, we demonstrate the practical value and robustness of the problem tailored orthogonal polynomial basis generation algorithm. In order to do this we generated rational approximations to data sets which are otherwise impossible to approximate using the monomial basis representation. We start by very informally discussing some computational aspects of the algorithms and then show examples of its application to complicated data sets.

A. Discussion of Complexity

Generating the approximations consists in two steps. In the first step we must compute the approximations coefficients, $\tilde{a}_k$, $\tilde{b}_k$ corresponding to (9) (or $\mathbf{A}_k$, $\mathbf{B}_k$ in (22) for the MIMO case). Then a state-space representation must be generated.

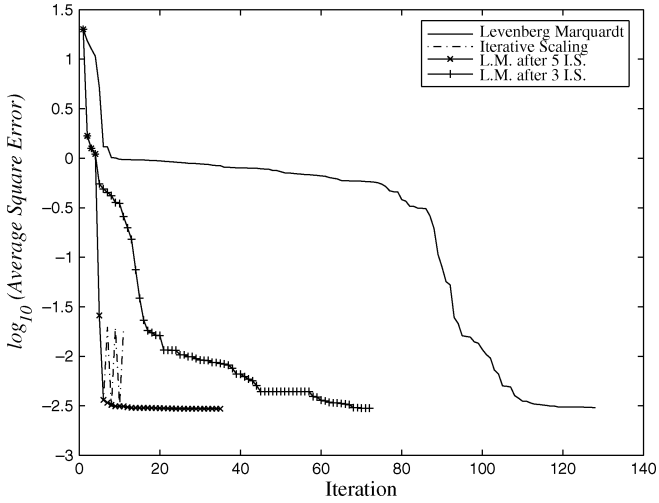

Fig. 2. Convergence curves for a SAW filter example.

This second step is trivially performed, as we have shown previously how to directly generate the $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ state-space representation from the orthogonal basis coefficients. The first step is accomplished by solving a nonlinear equation, such as that resulting from minimizing (3) in an appropriate norm. The nonlinear minimization algorithms used in this section are detailed in [15].

We will not dwell too much on the choice of nonlinear minimization algorithms, as the main point of this paper is the choice of the model representation and not the optimization algorithms themselves. Nevertheless, for the purpose of completness we point out that the procedure detailed is indeed very efficient. The setup of the equations is quite trivial and the only real computation corresponds to the nonlinear minimization described. In [15], a specific procedure involving the Levenberg–Marquardt algorithm is described to minimize $\|\tilde{\mathbf{E}}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, s)\|_2 = \|\tilde{\mathbf{R}}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, s) - \mathbf{H}(s)\|$. Also, each Levenberg–Marquardt iteration implies solving a linear system, which has a computational cost which is cubic in the number of unknowns (the number of coefficients required). Given the size of the approximations one is generally dealing with, on the order of up to a few hundred unknowns, this cost is acceptable if the Levenberg–Marquadt algorithm can be made to converge in a small number of iterations. The reason this fast convergence is achievable is that, as briefly hinted in Section III-C, we use the solution of a scaled linearized problem for generating a very good initial guess (this algorithm is termed Iterative Scaling). Therefore, the cost of modeling, for all of the test cases we have investigated was on the order of seconds for typical examples, to a small number of minutes at most, for the most complex models, having hundreds of states. Fig. 2 shows a plot of the square of the approximation error with respect to the number of iterations for a simple surface acoustic-wave (SAW) filter (details of this example are given below in Section V-B). Shown on the plot are the modified Levenberg–Marquardt (LM) method (using the orthogonal basis scheme we described), our basic iterative scaling linearized rational fitting (IS) method and also an hybrid rational fitting algorithm (Hybrid) that starts the nonlinear minimization after a few iterations of the linearized procedure. As can be seen from the figure the hybrid method outperforms the standard LM in speed, showing a much faster convergence, as well as the IS in robustness, thus allowing the efficient and robust generation of accurate high-order models.

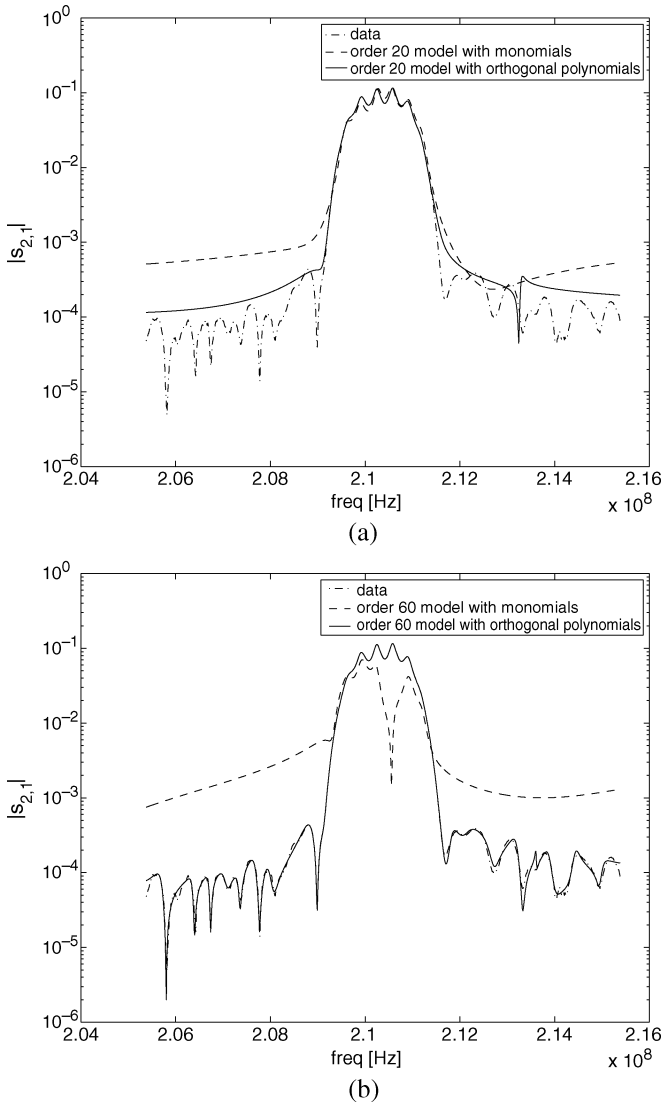

Fig. 3. Monomial-based and orthogonal-based approximations to the magnitude of an SAW filter s_{21} parameter using (a) 20th- and (b) 60th-order models.

B. Experimental Data

Our first example is the s_{21} parameter of a SAW filter.[1] Typically a SAW filter is a narrow band filter with very high sideband rejection and a large delay. Accurately capturing the phase associated with this delay already requires a very high-order approximation. After normalizing the frequency range, which spans from 60 to 80 MHz, the Vandermonde matrix loses rank after order 10, which makes it impossible for a standard fitting procedure to capture the delay.

Fig. 3(a) and (b) dramatically shows the limitations of the standard monomial basis approximation schemes. Here, sampled data and two rational approximations for the magnitude and phase of the s_{21} parameter of a SAW filter are shown. Fig. 3(a) shows that for order 20, the monomial representation is already considerably less accurate than the orthogonal basis

[1]Data provided courtesy of John Voll, Sawtek Inc.

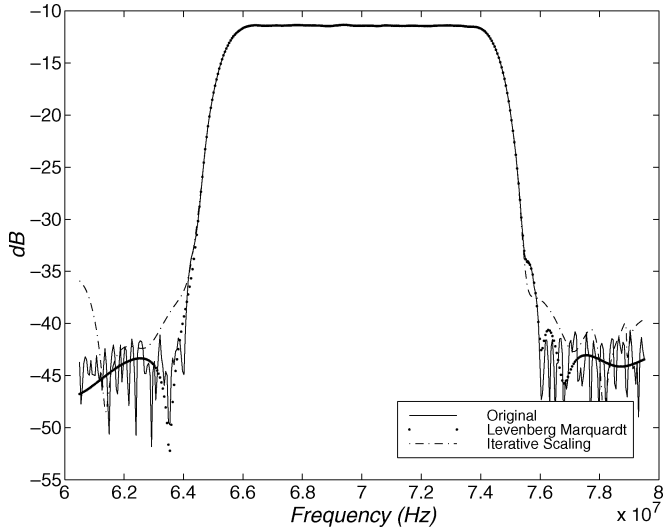

Fig. 4. SAW filter s_{21} approximation using 60th-order models (magnitude).

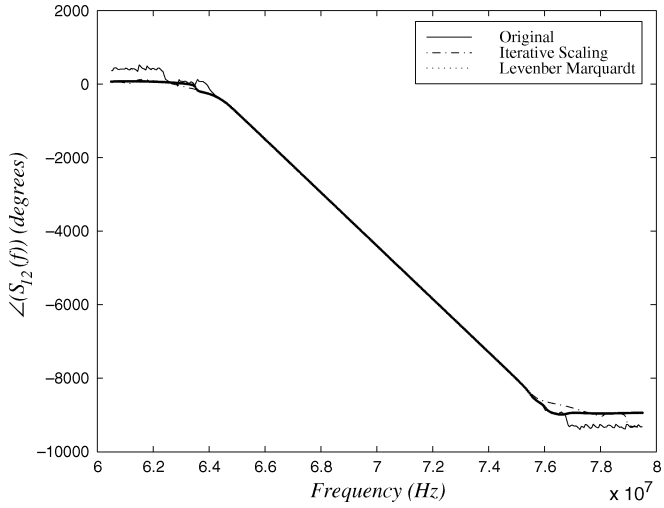

Fig. 5. SAW filter s_{21} approximation using 60th-order models (phase).

representation. The model generated with the proposed algorithm, on the other hand, is already quite accurate, being able to clearly distinguish the pass band and the rejection side bands. We note that even after scaling, the underlying least-squares matrix is so badly conditioned that no further improvement can be obtained (consider that entries in that matrix corresponde to values of magnitude 1 as well as frequencies raised to the 20th power). In fact, Fig. 3(b) shows that, increasingly, numerical random error is introduced in the procedure, and a 60th-order monomial-based approximation is actually less accurate than its lower order counterpart. The orthogonal representation however, shows no such difficulties and is able to approximate the original data with high accuracy.

In Figs. 4 and 5, we again show sampled data and two rational approximations for, respectively, the magnitude (now in the usual decibel representation) and phase of the s_{21} parameter of a SAW filter. The rational approximations were obtained using the modified Levenberg–Marquardt algorithm and the iterative scaling algorithm. Both of the approximations shown in the figures are of order 60 and both were stable. As can be seen from the figure, both approximations show high accuracy in representing the original data both in the narrow band as well as in capturing the sharp transition in frequency that is typical of such systems. The rejection side bands are also well approximated even though the magnitude of both the data and the models in that region is approaching numerical noise. It would be virtually impossible to attain similar accuracy with the low-order model restriction imposed by the use of the monomial basis.

For our next example, we simultaneously approximated the four (4) S parameters of a spiral inductor, modeled as a two-port. The model is composed of two single input, two output models. In this case the frequency data spans several decades, there is a high quality resonance and some high frequency detail. The Vandermonde matrix for this problem loses rank after order 11 and increasing model order beyond that point leads to no additional model accuracy. In order to fit the data with acceptable accuracy, we needed to at least go up to order 20 models for each sub-model. This would have been impossible using the monomial basis.

Fig. 6 shows plots of the magnitude of the four Y parameters. The curves shown are the original sampled data and the result of a 30th-order approximation. From the figures it is clear that the sharp zero is very accurately fitted, as well as the remaining frequency range, including the high frequency details. Again such fittings are next to impossible to obtain with low-order models. The figures shows that our MIMO formulation derived in Section IV can indeed be used to generate highly accurate models of sampled data for MIMO systems. While it is possible to approximate the system as four separate SISO systems, the resulting model would have size 4×30, which is very inefficient.

Our third example actually comes from a different setting. The data was obtained from a linear model of a component of the International Space Station. The model was previously described in [23] and is sometimes used as an benchmark for model order reduction algorithms. Here we use samples obtained from the model to demonstrate the generality of our proposed technique and in particular to show that it can be used to robustly generate high-order accurate models. The structure in question is a 3×3 MIMO model with a complicated frequency response and was approximated with a 65th-order model. Fig. 7 shows the original data and approximation to two representative entries in the matrix transfer function. The two curves are virtually indistinguishable.

Our final example shows the application of these models in a simulation environment. Here, we show the time-domain response of a transmission line driven by a pulse waveform. Transmission lines contain ideal delays that are difficult for rational approximants to capture accurately. For this reason, efficient transmission line simulation algorithms usually extract any ideal delay before applying rational approximation techniques. We have retained the delay in the frequency-domain s-parameter data in order to illustrate the ability of our algorithms to robustly generate high-order models. For the simulation parameters chosen, the rise time t_r was about 2.5 times the time of flight delay T_d along the transmission line. A large mismatch was deliberately introduced between the transmission line, driver, and load so that strong reflection effects could be observed. The rational approximation required an order 70, 2-input, 2-output model to match s-parameters up to a frequency of $12.5/T_d =$

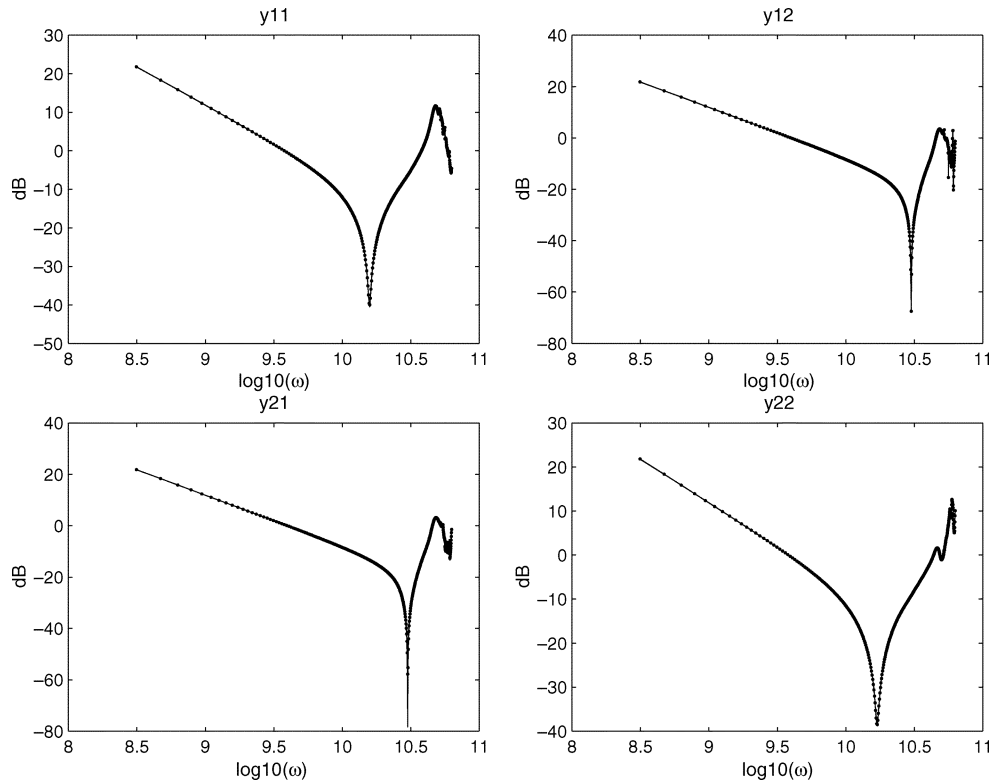

Fig. 6. Magnitude of spiral inductor Y parameter: original data (solid line) and approximation (dot markers).

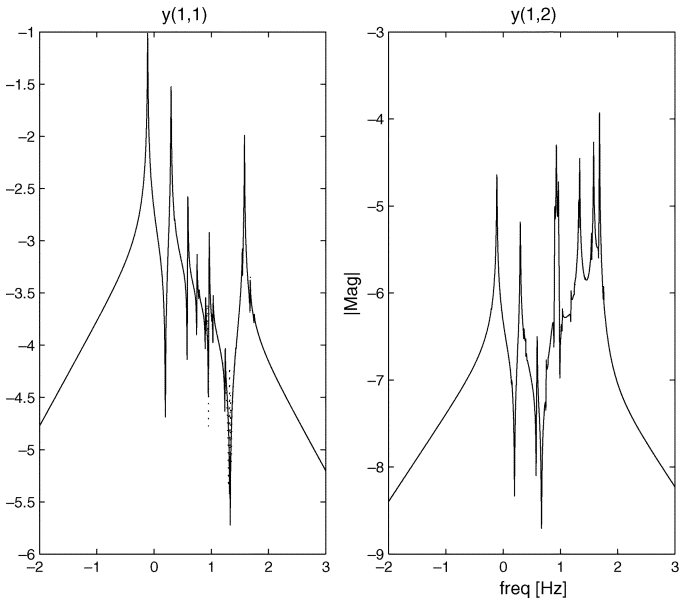

Fig. 7. Magnitude of structural model of component from the International Space Station: original data (solid line) and approximation (dot markers). The two plots are almost indistinguishable.

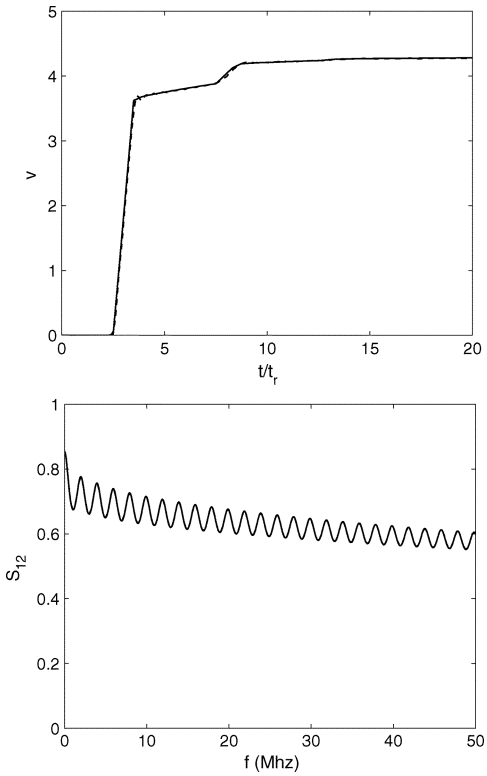

Fig. 8. End response of transmission line driven with 5-V step input waveform. Solid lines show full transmission line model, dashed lines show rational approximated model. Top: $|S_{12}|$. Bottom: transient response to step input with 50-Ω drivers and receivers.

$5/t_r$. A relative error tolerance of 1% and absolute error tolerance of 10^{-4} was used in the error weighting. Fig. 8 shows the response of the original and approximate transmission line model. Overall the agreement is excellent. A small amount of Gibbs-effect related ripple is barely noticeable near the first plateau in the waveform. This is due to the truncation of the data used for the rational approximation at a finite frequency.

VI. Conclusions and Future Work

In this paper, we presented a robust method for generating an orthogonal polynomial basis from tabular data in the frequency

domain. We also showed that the formulation used to generate the basis can be extended to directly produce a state-space model based on a generalized controller canonical form that can be used in time-domain simulation. We then demonstrated the value of generating problem specific orthogonal polynomials basis by calculating rational approximations of several data sets. The data sets contain enough detail that any attempt at approximating them using standard monomial basis would fail. This shows that high-order, high-accuracy state-space models can be obtained in a robust fashion. While the proposed method does not guarantee the stability or passivity of the approximation, it can readily be used as a highly accurate initial approximation for well-known passivity enforcing procedures. In this context, the proposed method can be seen as an efficient enabler or an essential first step for such methods.

REFERENCES

[1] L. M. Silveira, I. M. Elfadel, J. K. White, M. Chilukuri, and K. S. Kundert, "An efficient approach to transmission-line simulation using measured or tabulated S-parameter data," in *Proc. 31st Des. Autom. Conf.*, San Diego, CA, Jun. 1994, pp. 634–639.

[2] T. V. Nguyen, J. Li, and Z. Bai, "Dispersive coupled transmission line simulation using an adaptive block lanczos algorithm," in *Proc. Int. Custom Integr. Circuits Conf.*, May 1996, pp. 457–460.

[3] G. Zheng, Q.-J. Zhang, M. Nakhla, and R. Achar, "An efficient approach to moment-matching simulation of linear subnetworks with measured or tabulated data," in *Proc. Int. Conf. Comput. Des.*, San Jose, CA, Nov. 1996, pp. 20–23.

[4] A. Deutsch, G. V. Kopcsay, P. J. Restle, H. H. Smith, G. Katopis, W. D. Becker, P. W. Coteus, C. W. Surovic, B. J. Rubin, R. P. Dunne, T. Gallo, K. A. Jenkins, L. M. Terman, R. H. Dennard, G. A. Sai-Halasz, B. L. Krauter, and D. R. Knebel, "When are transmission line effects important for on-chip interconnections?," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 10, pt. 2, pp. 1836–1846, Oct. 1997.

[5] J. R. Phillips, E. Chiprout, and D. D. Ling, "Efficient full-wave electromagnetic analysis via model-order reduction of fast integral transforms," in *Proc. 33rd Des. Autom. Conf.*, Las Vegas, Nevada, Jun. 1996, pp. 377–382.

[6] W. T. Beyene, "Improving time-domain measurements with a network analyzer using a robust rational interpolation technique," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 3, pp. 500–508, Mar. 2001.

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

[8] D. K. de Vries and P. M. J. Van den Hof, "Frequency domain identification with generalized orthonormal basis functions," *IEEE Trans. Autom. Contr.*, vol. AC-43, no. 5, pp. 656–669, May 1998.

[9] B. Mihaila and I. Mihaila, "Numerical approximations using chebyshev polynomial expansions: El-gendi's method revisited," *J. Phys. A,* vol. 35, no. 3, pp. 731–746, 2002 [Online]. Available: http://stacks.iop.org/0305-4470/35/731

[10] L. M. Silveira, I. M. Elfadel, J. K. White, M. Chilukura, and K. S. Kundert, "Efficient frequency-domain modeling and circuit simulation of transmission lines," *IEEE Trans. Compon., Packag., Manufact. Technol, B, Adv. Packag.*, vol. 17, no. 4, pp. 505–513, Nov. 1994.

[11] S.-H. Min and M. Swaminathan, "Construction of broadband passive macromodels from frequency data for simulation of distributed interconnect networks," *IEEE Trans. Electromagn. Compat.*, vol. 46, no. 4, pp. 544–558, Nov. 2004.

[12] J. Morsey and A. C. Cangellaris, "PRIME: Passive realization of interconnect models from measured data," in *Proc. IEEE 10th Topical Meeting Elect. Perform. Electronic Packag.*, Oct. 2001, pp. 47–50.

[13] 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.

[14] C. P. Coelho, J. R. Phillips, and L. M. Silveira, "A convex programming approach to generating guaranteed passive approximations for tabulated frequency-data," *IEEE Trans. Comput.-Aided Design*, vol. 23, no. 2, pp. 293–301, Feb. 2004.

[15] ——, "Robust rational function approximation algorithm for model generation," in *Proc. 36th Des. Autom. Conf.*, New Orleans, LA, Jun. 1999, pp. 207–212.

[16] G. H. Golub and C. F. V. Loan, *Matrix Computations*, ser. Series in the Mathematical Sciences, 3rd ed. Baltimore, MD: The John Hopkins Univ. Press, 1996.

[17] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Boston, MA: PWS Publishing, 1996.

[18] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. Philadelphia, PA: SIAM, 1999.

[19] A. Antoniou, *Digital Filters Analysis, Design and Applications*, 2nd ed. New York: McGraw-Hill , 1993.

[20] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their l_∞ error bounds," *International J. Control*, vol. 36, pp. 1115–1193, 1984.

[21] T. Kailath, *Linear Systems*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[22] R. S. Varga, *Matrix Iterative Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1962.

[23] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, "A survey of model reduction methods for large-scale systems," in *Contemporary Mathematics*, V. Olshevsky, Ed. Providence, RI: AMS, 2001, vol. 280, pp. 193–219.

Carlos Pinto Coelho received the Engineer's and Master's degrees in electrical and computer engineering from Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, in 1999 and 2001, respectively. He is currently working toward the Ph.D. degree in the Computational Prototyping Group, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge.

His research interests are in the application of numerical simulation techniques to multiphysics problems arising in the context of bioengineering applications.

Joel R. Phillips (S'91–M'97–SM'05) received the Bachelor's, Master's, and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1991, 1993, and 1997, respectively.

He joined Cadence Design Systems, Berkeley, CA, in 1997, and is currently a Research Scientist with Cadence Berkeley Laboratories, Berkeley, CA. His research interests include the application of numerical simulation techniques to problems in electronic design automation. He received notable paper citations at DAC 2000, DAC 2002, SBCCI2002, and the 1995 Copper Mountain Conference on Multigrid Methods.

Luís Miguel Silveira (S'85–M'95–SM'00) was born in Lisbon, Portugal. He received the Engineer's (*summa cum laude*) and Master's degrees in electrical and computer engineering from Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, in 1986 and 1989, and the M.S., E.E., and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1990, 1991, and 1994, respectively.

He is currently a Full Professor of Electrical and Computer Engineering at Instituto Superior Técnico, Technical University of Lisbon, a Senior Researcher at Instituto de Engenharia de Sistemas e Computadores: Investigação e Desenvolvimento em Lisboa (INESC-ID), Systems and Computer Engineering Institute R&D, and a founding member of the Lisbon Center of the Cadence Laboratories. His research interests are in various aspects of computer-aided design of integrated circuits with emphasis on interconnect modeling and simulation, parallel computer algorithms and the theoretical and practical issues concerning numerical simulation methods for circuit design problems.

Mr. Silveira is a member of the IEEE and Sigma Xi.