Multi-Dimensional Automatic Sampling Schemes for Multi-Point Modeling Methodologies

Jorge Fernández Villena, *Member, IEEE,* and Luís Miguel Silveira, *Senior Member, IEEE*

***Abstract*— This paper presents a methodology for optimizing sample point selection in the context of model order reduction (MOR). The procedure iteratively selects samples from a large candidate set in order to identify a projection subspace that accurately captures system behavior. Samples are selected in an efficient and automatic manner based on their relevance measured through an error estimator. Projection vectors are computed only for the best samples according to the given criteria, thus minimizing the number of expensive solves. The scheme makes no prior assumptions on the system behavior, is general, and valid for single and multiple dimensions, with applicability on linear and parameterized MOR methodologies. The proposed approach is integrated into a multi-point MOR algorithm, with automatic sample and order selection based on a transfer function error estimation. Different implementations and improvements are proposed, and a wide range of results on a variety of industrial examples demonstrate the accuracy and robustness of the methodology.**

***Index Terms*—Automatic multi-dimensional sample selection, multi-point model order reduction, parameterized systems.**

I. Introduction

USING currently available tools in the EDA industry, the detailed models representing physical devices obtained after the modeling and extraction steps are often too large for practical or efficient simulation and verification. Reducing the complexity of these models, while guaranteeing input–output accuracy, is crucial to enabling the simulation and verification of those large systems [3], [4]. This is the realm of model order reduction (MOR) [5]–[7].

Multi-point based MOR approaches [8]–[11] have recently gathered renewed attention due to their robustness and reliability. In particular, some [12] have been extended to tackle the problem of parameterized model order reduction (pMOR), where system behavior depends on frequency and a set of parameters, modeling for instance process variations or environmental conditions. In this context, the multi-dimensional sample-based techniques seem less sensitive to the number of parameters, and become a good alternative to the multi-dimensional moment matching approaches [13]–[15], which seem unable to deal with what is commonly denoted as the curse of dimensionality (leading to oversized models when the number of parameters and the accuracy require matching a large set of moments). However, multi-dimensional sampling can be expensive, specially if there is no good indication of where to place the sample points. Random based sampling can defeat the reliability of these algorithms, whereas trying to cover the complete subspace with a linear scheme is overwhelming beyond a couple of dimensions.

In this paper we will try to fill this gap, by proposing a novel methodology for sample selection in single and multi-dimensional spaces. The procedure, which is directly related to (but not dependent upon) [9], is based on maximizing the subspace spawned with the sampling set, in order to obtain a good approximation of the dominant subspace in the orthonormalization step. To avoid falling into inefficiency, the algorithm will iteratively select a good new sample point without solving the system, and assuming no prior knowledge of the system. System solution is only performed for the selected samples. The method will also avoid oversampling by estimating the error on the fly. It will decide at run time whether to generate a new sample, and where to place it. Therefore, the proposed methodology will present a fully automated procedure, which will try to overcome the lack of automation in existing (sample-based) MOR procedures, and in particular in the pMOR scenario. Although beyond the scope of this paper, the procedure is general enough to be applied to different fields, such as non-linear MOR where a non-linear dependence with parameters or time is exhibited [3], [10], [16].

This paper is structured as follows. In Section II, an overview of the MOR paradigm is presented, along with a discussion of existing sampling schemes. In Section III, the theoretical basis for the sampling scheme is introduced. Section IV discusses how to cope with some of the practical issues arising from the theoretical concepts. In Section V, different efficient implementations of complete automated multi-dimensional MOR methods, which address the multiple practical issues of Section IV, are proposed. In Section VI, the efficiency and robustness of the proposed methods are shown

Manuscript received July 12, 2010; revised November 20, 2010; accepted January 6, 2011. Date of current version July 20, 2011. Preliminary results appeared in [1] and [2]. This work was supported in part by the EU IST CHAMELEON RF Project (FP6/2004/IST/4-027378), by the Fundação para Ciencia e Técnologia (FCT) (INESC-ID multiannual funding) through the PIDDAC Program funds, by FCT Grant SFRH/BD/61213/2009, and by NXP Semiconductors under EMonIC Project. This paper was recommended by Associate Editor P. Li.

J. F. Villena is with the Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa (INESC ID), Lisbon 1000-029, Portugal (e-mail: jorge.fernandez@inesc-id.pt).

L. M. Silveira is with Instituto Superior Técnico (IST), Technical University of Lisbon, Lisbon 1049-001, Portugal, and also with the Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa (INESC ID), Lisbon 1000-029, Portugal (e-mail: lms@inesc-id.pt).

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

Digital Object Identifier 10.1109/TCAD.2011.2158721

on a wide range of industrial examples. We will close with Section VII, where some conclusions are drawn.

II. Background

A. Parameterized Systems and Model Order Reduction

The main techniques in MOR are geared toward the reduction of a state space linear time-invariant representation of a physical system. In such representation, the output y is related to the input u via some inner states x. When parametric variations are taken into account, the system is represented as a parameterized state-space descriptor

$$\begin{aligned}(G(\lambda) + sC(\lambda))\, x(s, \lambda) &= Bu(s)\\ y(s, \lambda) &= Ex(s, \lambda)\end{aligned} \tag{1}$$

where $C, G \in \mathbb{R}^{n\times n}$ are, respectively, the dynamic and static matrices, $B \in \mathbb{R}^{n\times m}$ is the matrix that relates the input vector $u \in \mathbb{C}^m$ to the inner states $x \in \mathbb{C}^n$, and $E \in \mathbb{R}^{p\times n}$ is the matrix that links those inner states to the outputs $y \in \mathbb{C}^p$. We assume here, as is common, that the elements of C and G, as well as the states x, depend on a set of P parameters $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_P]^T \in \mathbb{R}^P$ which model the effects of the uncertainty. Usually, the system is formulated so that the input (B) and output (E) matrices do not depend on the parameters. Furthermore, for electrical-based systems, most of the modeling methodologies treat the inputs and outputs as bidirectional ports (i.e., every port has dual input and output quantities associated), and thus $B = E^T$. For the remainder of this paper, and without lack of generality, we will assume that this is the case. When this condition is not verified, a simple trick is to expand the input and output matrices, so that the new matrices are $\bar{B} = [B\ E^T] = \bar{E}^T$, with inputs and outputs $\bar{u} = [I\ 0]^T u$ and $y = [0\ I]\bar{y}$ (with I the identity and 0 a block zero matrix of commensurate dimensions).

Parametric dependence is often obtained via the sensitivity computation of the discretized elements with respect to the parameters [17]–[19]. Therefore, matrices C and G in (1) can be represented in a polynomial form via a Taylor series expansion with respect to the parameters

$$\begin{aligned}G(\lambda) &= \textstyle\sum_{\phi_1\ldots\phi_P=0\ldots0}^{k_1\ldots k_P} \Lambda_{\phi_1\ldots\phi_P} G_{\phi_1\ldots\phi_P}\\ C(\lambda) &= \textstyle\sum_{\phi_1\ldots\phi_P=0\ldots0}^{k_1\ldots k_P} \Lambda_{\phi_1\ldots\phi_P} C_{\phi_1\ldots\phi_P}\end{aligned} \tag{2}$$

where the matrices $G_{\phi_1\ldots\phi_P}$ and $C_{\phi_1\ldots\phi_P}$ (with ϕ_i positive integers) are the joint sensitivities of order $(\phi_1 \ldots \phi_P)$ w.r.t. to the P parameters, and the scalar $\Lambda_{\phi_1\ldots\phi_P} = \lambda_1^{\phi_1}\lambda_2^{\phi_2}\ldots\lambda_P^{\phi_P}$ is the generalized multi-parameter. The nominal matrices in this formulation would be $G_{0\ldots0}$ and $C_{0\ldots0}$. The Taylor series can be extended up to the desired (or required) order $k = k_1 + \ldots + k_P$, including cross terms, for the sake of accuracy. This system in (1) has an associated frequency response modeled via the transfer function

$$H(s, \lambda) = B^T \left(sC(\lambda) + G(\lambda)\right)^{-1} B. \tag{3}$$

The objective of pMOR techniques is to generate a reduced order approximation of (3), able to accurately capture the input–output behavior of the system for any point in the joint frequency-parameter space

$$\widehat{H}(s, \lambda) = \widehat{B}^T \left(s\widehat{C}(\lambda) + \widehat{G}(\lambda)\right)^{-1} \widehat{B} \tag{4}$$

where $\widehat{C}, \widehat{G} \in \mathbb{R}^{q\times q}$ and $\widehat{B} \in \mathbb{R}^{q\times m}$ are the reduced set of matrices, with $q \ll n$ the reduced order.

In general, one attempts to generate a reduced order model (ROM) whose structure is as similar to the original as possible, i.e., exhibiting a similar parametric dependence allowing more control within analysis and optimization frameworks, in order to facilitate further simulations.

B. Projection Based Approaches

The most common procedure to obtain an accurate and structurally similar ROM is to combine the sensitivity-based Taylor series representation from (2) with an orthogonal projection scheme, as presented in [13].

Standard pMOR methodologies rely on the generation of a suitable low order subspace (spanned by a basis $V \in \mathbb{R}^{n\times q}$), where the original system matrices $C(\lambda)$, $G(\lambda)$ and B are projected, and the resulting reduced model (4) captures the behavior of the system under parameter variations

$$\begin{aligned}\widehat{C}(\lambda) &= V^T \left(\textstyle\sum_{\phi_1\ldots\phi_P=0\ldots0}^{k_1\ldots k_P} \Lambda_{\phi_1\ldots\phi_P} C_{\phi_1\ldots\phi_P}\right) V\\ \widehat{G}(\lambda) &= V^T \left(\textstyle\sum_{\phi_1\ldots\phi_P=0\ldots0}^{k_1\ldots k_P} \Lambda_{\phi_1\ldots\phi_P} G_{\phi_1\ldots\phi_P}\right) V\\ \widehat{B} &= V^T B \qquad x(s, \lambda) = V\widehat{x}(s, \lambda)\end{aligned} \tag{5}$$

where $V \in \mathbb{R}^{n\times q}$ spans the projection subspace of reduced dimension q, and $\widehat{C}, \widehat{G} \in \mathbb{R}^{q\times q}$, $\widehat{B} = \widehat{E}^T \in \mathbb{R}^{q\times m}$, and $\widehat{x} \in \mathbb{C}^q$ define the ROM.

To ensure the accuracy of the ROM, the basis $V \in \mathbb{R}^{n\times q}$ must be able to span the solution of $x(s, \lambda)$ for the relevant region Γ of the $\{s, \lambda\}$ space, or in other words

$$x(s, \lambda) \approx \textstyle\sum_{i=1}^{q} \alpha_i(s, \lambda) V_i \quad \forall\ \{s, \lambda\} \in \Gamma \tag{6}$$

where V_i is the ith column of the projector V, and $\alpha_i \in \mathbb{C}$.

Multiple approaches have been presented in order to generate the matrix V. Most of the techniques in the literature extend the moment matching paradigm [6] to the multi-dimensional case [13]–[15]. They usually rely on the implicit or explicit matching of the moments of the parametric transfer function (3). Following the same idea used in the nominal moment matching techniques, a basis for the subspace can be built from the column span of such moments, and the resulting matrix V can be used as a projection matrix so that the ROM implicitly matches the selected moments. Approaches differ in which moments are matched and how these moments are generated (for a more complete review, the reader is referred to [13]–[15]). In general, these methods, which rely on local matching, suffer from oversize of the models when the number of moments to match is high, either because high order is required, or because the number of parameters is large. A different paradigm, more appealing to our goals, is based on multi-point methods, which generate the basis either by computing the transfer function moments from multiple expansion points (s_k, λ_k), or from solving the system at different sample points on the relevant frequency plus parameter space

$$z_k = z(s_k, \lambda_k) = (s_k C(\lambda_k) + G(\lambda_k))^{-1} B \tag{7}$$

where $z(s_k, \lambda_k) \in \mathbb{C}^{n\times m}$ is the block vector generated at the sample point (s_k, λ_k), and related to the zeroth order (block) moment. In this framework, the approach in [12] presents a statistical interpretation of the algorithm in [9], and enhances its applicability. This approach also appears to be less sensitive to the number of parameters. In addition, the accuracy does not depend on the "quality" of the quadrature of the integral representation of the Gramian, but on the approximation of the subspace [9]. Regrettably, very little automation has been reported on pMOR methodologies, since the number and placement of samples must be decided beforehand. Bad or poor sampling may lead to inaccurate results (undersampling), whereas an excessive number of samples can lead to inefficiency (oversampling), and in some cases both effects can happen in different regions of the space. Even worst is the fact that there is no efficient method to determine if the reduction suffers from either or both effects, which leads to a small level on confidence on the ROM.

C. Sample Selection Schemes

Little work has been devoted to the issue of sample selection inside the framework of projection-based MOR. The authors in [20] presented an iterative procedure for selecting the r best interpolation points that generate an optimal $\|H\|_2$ model of order r. The algorithm generates r initial random samples, used to obtain a rth order model. The eigenvalues of this reduced model are obtained (this operation is relatively cheap on the reduced model), and used as a new set of r sampling points. The procedure is repeated until the eigenvalues (i.e., next iteration sampling points) converge, generating the $\|H\|_2$ optimal model. Obvious drawbacks to this procedure are that it is only applicable to single dimensional linear systems, and requires solving the original system (an expensive operation) at r points on each iteration. Also, increasing r (the reduced order, which has a dramatic impact on accuracy and is not known beforehand) implies complete model recalculation, and the initial guess of the points is relevant to the convergence of the eigenvalues. This means that some *a priori* knowledge is needed for a good initial guess.

The work presented in [21] is explicitly aimed at sample point selection for optimizing multi-point MOR schemes. There, the idea is to generate R ROMs of identical reduced size q, resulting from different sample sets, $Z_1 \ldots Z_R$, obtained by bootstrapping from a larger common set of samples. When comparing their response, regions where associated transfer functions $H_k(s)$ (with $k = 1 \ldots R$) disagree the most are likely to be good choices to place new sample points (see [21] for details). The method also has some drawbacks: one is that it requires evaluating multiple reduced models' frequency transfer functions, and another is that it requires computing an initial population, i.e., an initial set of samples (and vectors) to start working with (and with no guarantee that these first samples are a good set). These issues are aggravated in multi-dimensional spaces: how to compute efficiently the multi-dimensional variance, and how to set a first sample population in this case?

III. Automatic Sampling Selection

This section introduces the proposed sampling scheme, which is based on trying to find the minimum set of samples that will spawn a subspace that generates a good ROM.

A. Motivation and Basic Concepts

Before presenting our proposed procedure, we have to settle on two basic premises. If we define $A_j = s_j C(\lambda_j) + G(\lambda_j)$, $A_j \in \mathbb{C}^{n\times n}$, the evaluated system matrix for the jth sampling point (s_j, λ_j), we note the following.

1) *Evaluation of the system matrix is cheap*. It requires the evaluation of the entries of A_j. This could be the case when the matrix is a known function of the frequency and parameters, or can be represented as a polynomial function of the frequency and parameters.
2) *Solving the system is expensive*. We have to obtain the sample block vector $z(s_j, \lambda_j) = A_j^{-1} B$, which requires solving the system, either by a direct method [22] (e.g., LU factorization of A_j, plus backsolves), or by an iterative method [23]. In either case, this operation can be expensive for large systems.

This means that we should avoid computing the sample block vectors $z(s_j, \lambda_j)$ unless it is strictly necessary, i.e., that we are sure we need the vector. Any sample may give a good vector for approximating the system, at least in the neighborhood of the point. On the other hand, a new vector may add information that is already enclosed in the subspace generated so far (i.e., redundant information), and thus it is not a good new sampling choice. Therefore, a good new vector is the one that is as different from the ones we already have as possible, and thus, cannot be well approximated by the currently available subspace.

However, this unleashes the question: how can we determine if a candidate sample point (s_j, λ_j) will generate a block vector that adds rank to our set $Z = \{z_1 \cdots z_k\}$ without computing it? Furthermore, how can we know if this new block vector will help to minimize the number of samples needed to obtain a good ROM? The answer is simple, we cannot. But on the other hand, we can do some simple and efficient computations that will provide a good estimate of the best candidate sample point to solve for.

B. Iterative Sample Selection Methodology

Our goal is to obtain a minimum number of vectors so that we can obtain a good approximation of the states vector, i.e., we want to find the minimum set q so that (6) holds. The vectors are going to be generated via a multi-point sample scheme, and we want to minimize the number of samples generated (i.e., the number of solves to perform).

We start from an initial candidate set of T points, $\Psi = \{\psi_1 \ldots \psi_T\}$, covering the space of interest, Γ, and where each $\psi_j = (s_j, \lambda_j)$ is a point in that space Γ. We will select from

Ψ the most appropriate point for our goals by trying to find which points have an associated vector that is less similar to the vectors we have already computed. It is important to notice that we want to obtain this information before solving the system for the candidate point, and assuming no prior knowledge of the system behavior is available.

Let us assume that at certain point during our algorithm, we have a basis $V \in \mathbb{R}^{n\times k}$, $V = [v_1 \dots v_k]$. For simplicity, let us assume a single input system, i.e., the state space response is due to the contribution of a single input, and thus B is a column vector $b \in \mathbb{R}^n$. The error we commit in the approximation of the state vector at a given point ψ_j is

$$e_j = x(\psi_j) - \textstyle\sum_{i=1}^{k} \alpha_i(\psi_j) v_i \tag{8}$$

where $e_j \in \mathbb{C}^n$ is the error, and $\alpha_i \in \mathbb{C}$. This error gives us true information, and thus a good option is to solve the system at any candidate sample point in which the error is larger. However, to generate the error we would need to compute the solution of the system at each candidate $x(\psi_j)$, which is exactly what we want to avoid.

Instead of the true error we can use a proxy that is cheaper to compute and that is a good indicator in the case of linear systems: the residue. Multiplying by the system matrix evaluated in this point, $A_j = A(\psi_j)$, we know that $b = A_j x(\psi_j)$, and if we denote $r_j = A_j e_j$, $r_j \in \mathbb{C}^n$, we obtain

$$r_j = b - \textstyle\sum_{i=1}^{k} \alpha_i(\psi_j) A_j v_i. \tag{9}$$

Therefore, to determine whether the system $\{A_j, b\}$ is well approximated by basis V, we simply orthogonalize the vector b against the basis $A_j V$, and generate the norm

$$\|r_j\| = \|b \perp A_j V\| \tag{10}$$

where r_j is the residue after the orthogonalization ($\perp$) of b against the set of vectors $A_j V$. In the case of a MIMO system, since the state vector is obtained as the contribution of the multiple inputs, and thus related to the m columns of B, the overall residue norm can be obtained as the contribution of the norms of the residues for each of the columns of B (nonetheless, the residue can be obtained as a block, and the norm obtained as the sum of the column norms)

$$R = B \perp A_j V, \quad \|r_j\| = \textstyle\sum_{i=1}^{m} \|R_i\| \tag{11}$$

where $R \in \mathbb{C}^{n\times m}$, and R_i is the ith column. If the resulting norm $\|r_j\|$ is small, the system $\{A_j, B\}$ is well approximated by the set of vectors V, and thus, the vector associated with the candidate point ψ_j will probably add "small" rank to the subspace spanned by the set of vectors V. We can repeat the operation in (11) for all the candidate sample points to know for which of them the solution is poorly approximated by the set of vectors. This information can be obtained by finding the maximum among the norm of the residues given by (11) for each candidate point j. With this methodology we ensure that the new vector generated by the selected sample point adds rank to the subspace, and furthermore, it is likely to be a relevant point to include in the sample set.

Only after we have selected the best suited candidate ψ_j, do we solve the system to generate the block vector z_j, and withdraw the sample point from the candidate set, $\Psi = \{\psi_1 \dots \psi_{j-1} \quad \psi_{j+1} \dots \psi_T\}$. Then we repeat the procedure for finding the next point.

IV. Computational Issues

A. Computational Cost

The cost of standard sample based procedures can be measured as the contribution of two major factors. The first is the cost of generating a sample, that is, the cost associated with evaluating and solving the system for the sample point. This cost depends on the matrix profile (number of non-zeros and sparsity pattern), and the underlying algebra applied. Therefore, we define this cost as $O(S)$. The second factor is the cost of orthonormalizing a set of q vectors of size n, and can be approximated by $O(nq^2)$.

For the proposed methodology, let us suppose we have an initial number T of candidate points, and we determine a final number of K sample points, which generate a total number of Km vectors, with m the number of ports. In this case, we have to perform K solves with cost $O(KS)$ and the orthonormalization of the Km vectors, with cost $O(n(Km)^2)$. In addition, we have a major extra cost that comes from the set of operations associated with (11), required for finding the points to solve. This cost can be approximated by the expression

$$O(\textstyle\sum_{i=1}^{K} (T - i) n (im)^2). \tag{12}$$

The reasoning is that for finding the new sample point, at each remaining candidate point (a total number of $T - i$, with i the number of samples already computed), we must orthogonalize the columns of $B \in \mathbb{R}^{n\times m}$ against the set of vectors $A_j V \in \mathbb{C}^{n\times im}$ (recall that the basis at this point is $V \in \mathbb{R}^{n\times im}$). This cost can be further approximated by

$$O(nm^2 K^3 (T - K)). \tag{13}$$

Taking all into account, the total cost of the algorithm can be approximated by

$$O(KS + n(Km)^2 + nm^2 K^3 (T - K)). \tag{14}$$

We realize that the cost of the algorithm is highly dependent on the number of iterations K needed to accurately model the system, and the number T of initial samples on the candidate set. This cost will be taken into account in future sections to present efficient versions of the algorithm that minimize the impact of T, and thus the additional cost. In addition the method provides other advantages in terms of accuracy, reliability and automation that may compensate the extra cost. Furthermore, there is an enormous potential for parallelism, since the residue computation [responsible for the later terms in (14)] is embarrassingly parallel. Such a study is, however, beyond the scope of this paper.

B. Stopping Criteria

An important issue is when to stop looking for new samples, this is, when to stop the procedure. The answer is simple: one should stop when the solution of the remaining candidates (i.e., the vectors that can be generated) does not add rank. This means that the remaining points would only add redundant information to our basis. The proposed methodology naturally provides different indicators that can be used to monitor this effect. One is the norm of the residue. If this is a good proxy of the error, we can determine that the solution of the states is well approximated when the norm of all the residues is sufficiently small. However, when generating and comparing the residue it is important to take into account the nature of the variables provided by the formulation: different variables should be treated separately, and the measure of the residue should be relative to the norm of B (or sum of column norms). In addition, this information is only given for the remaining candidates, and thus a bad or incomplete set of initial candidates may lead to a wrong conclusion. The other is the norm of the new vector when orthogonalized against the current basis. An efficient method for obtaining such information is to use an incremental QR decomposition, for example a rank revealing (RR) QR [24]. Every time a new sample vector is generated, we orthonormalize it against the already orthonormalized set of vectors, and check its norm against a fixed tolerance (either absolute or relative). If the norm drops below such tolerance, we can determine that the vector does not incorporate relevant information (i.e., the information is already enclosed in the basis).

Both indicators can be combined to improve the reliability: when the norm of the residues falls below a fixed threshold based on a relative tolerance, it means that all the candidate points are well approximated. At this point, the norm of the orthogonalized vectors can be compared against a tolerance and used as a stopping criteria. A key advantage of the proposed approach relies in the confidence level on the ROM accuracy. When the stopping criteria are achieved, we can be sure that a good approximation of the model has been generated (as long as the initial candidate set is representative), contrary to the lack of reliability of existing pMOR approaches.

Although a theoretical error bound is not (yet) available, the provided monitoring is completely integrated in the methodology and can be used as a good approximation. In practice, we have reason to believe it is a reliable indicator.

C. Initial Candidate Set

A natural question arising from the previous sections is how to select an initial candidate set Ψ. The number of candidates will have a critical impact on the cost of the methodology, as presented in (14), and thus we are not interested in having an extremely fine discretization of the space under study. Uniform sampling in large dimensional spaces can be very expensive, whereas pure-random sampling can lead to clustered information in some regions with no samples in other regions.

One of the underlying ideas of the proposed method is to generate vectors as different as possible, so that the basis converges to the target subspace as fast as possible. This can be applied to the initial candidate set.

In order to improve the coverage of this initial set, a logarithmic mesh is defined in the complex frequency, and for each frequency point different parameter perturbations are performed following a low-discrepancy sequence [25]. The reason for the differentiated treatment for frequency comes from the fact that the range of variation of frequency is much wider, and frequency has in general, by design, a larger impact in the performance of electric circuits than the remaining parameter perturbations. Low-discrepancy or quasi-random sequences are routinely used in numerical integration of large dimensional problems. Since we are doing a numerical quadrature in a large dimensional space, the points generated by such schemes provide a better alternative to linear or pure random schemes, and ensure that each sample is different. Other schemes are, of course, possible.

Another advantage of the methodology is the flexibility. At each iteration different candidate sets can be used, or the initial candidate set can be refined on the fly during the procedure.

Although in this manuscript we have focused our attention on a static and relatively fine discretization of the domain of interest rating accuracy over efficiency, other strategies are also applicable, which opens the discussion to a large set of potential configurations: one could apply independently the procedure on disjoint space regions; apply different levels of accuracy in different regions; perform an initial approximation with a coarse candidate set and upgrade the model with a refined candidate in a second stage; use statistical or other information (if available) to generate a more realistic or better distribution of samples; monitor the selected samples and refine the candidate set on the fly in regions that demands multiple samples (since they are prone to have a lot of dynamics); perform a first approximation with frequency samples, and then incorporate the parameter samples at a second stage, with a finer discretization on regions with resonances (the parameters are prone to modify the amplitude or shift these resonances); and so on. The possibilities are endless, but their investigation is beyond the scope of this paper.

D. Stability and Passivity

The preservation of passivity and stability is a very important concern in MOR, in particular for linear and parameterized systems. With respect to this, we note that the methodology applies a congruence transformation on the original system, and thus the stability of the original model is preserved (of course as long as the original model is stable). Furthermore, under certain numerical and structural properties of the matrices, namely $C, G \geq 0$ (i.e., positive definite) and $B = E^T$, the passivity of the original system is also preserved (see [6] for details).

V. Proposed Implementation

In this section we will present different implementations, starting from a standard approach, and several upgrades in order to cope with some of the issues mentioned in the previous sections.

A. ARMS: Automatic Residue-Minimization Sampling

We start by presenting an approach whereby the selection of the samples is done by applying the procedure introduced in Section III-B. An initial population is defined as a static candidate set, and a first sample from this population is solved. This

Algorithm 1 ARMS: sorted residues norm update

```
Given the system and the domain of interest,
 1: Define stopping thresholds for the residue $t_r$ and the vectors $t_v$
 2: Generate a set of candidate sample points $\Psi = \{\psi_1 \dots \psi_T\}$
 3: Initialize an array with the residues $R(i) = 2\,\|B\|$
 4: Initialize: $V$=[ ], $k$=0
 5: Evaluate system matrix: $A_k$=$A(\psi_k)$
 6: Solve the system: $z_k$=$A_k^{-1}B$
 7: Orthonormalization: $v_k$= $z_k - V(V^T z_k)$, $N_v = \|v_k\|$, $k$++
        IF $N_v > t_v$   Then $V = \left[V\ v_k/N_v\right]$,
 8: Set the value $R_{max}$=0
 9: FOR $i$=$k$ : $T$
        $A_i$=$A(\psi_i)$, $r_i$=$B\perp A_i V$, $R(i) = \|r_i\|$
        IF $R(i) > R_{max}$   Then   $R_{max} = R(i)$
        IF $R_{max} > R(i$+1$)$   Then   BREAK
10: Sort $R$ in decreasing order, and $\Psi$ accordingly
11: IF $R(k) > t_r$ and $N_v > t_v$
        GOTO 4
12: Use $V$ in a congruence projection on the system
```

can be randomly selected, but it is a good idea to include the DC point (and thus obtain the DC moment), which for circuit simulation is quite relevant. The point chosen is removed from the candidate set. Once we have this vector, we orthonormalize it, and the result forms the current basis. With this basis we perform the operation in (11) to generate the residues at each candidate point. From these residues, we select the point with maximum residue norm, and solve it. The vector generated is orthogonalized against the previous ones (via an incremental RRQR). The norm of the orthogonalized vector and the residue is compared against a (relative) tolerance to determine if the procedure should end. If so, we terminate, and apply the basis in a congruence projection; if not, the sample is removed, the vector is added to the basis, and a new iteration is performed. An important point to mention is that if the sample is complex, then the vectors are also complex. As a consequence, in order to span the same column subspace we must add both the real and imaginary parts of the vectors to the basis (which must remain real).

B. ARMS: Sorted Residues Improvement

As pointed out, the main overhead of the method comes from the orthogonalization required in (11) for finding the residue at each candidate point. In the algorithm presented in the previous section, this operation was repeated for each candidate point at each iteration. However, in this section we will present an approach that will avoid most of such computations. If we have a basis $V \in \mathbb{R}^{n\times k}$, and an extra vector v_{k+1}, it is straightforward that

$$\left\|b\perp A_j V\right\| \geq \left\|b\perp A_j[V\ v_{k+1}]\right\|. \tag{15}$$

We can use this result to stop the search for the new sample point before recomputing all the residues.

Let us suppose that at a given iteration k, we have computed all the residues with the current basis $V_k \in \mathbb{R}^{n\times k}$, and we keep the norm of such residues, associated with each point, in a sorted array R, so that its entries are $\dots \geq \left\|r^k\right\|_j \geq \left\|r^k\right\|_{j+1} \geq \dots$, where superscript k indicates iteration, and subscript j indicates array index. In order to keep track of which residue belongs to which point, the array with the samples must be sorted accordingly (or, for simplicity, an incidence array is kept).

For the next iteration we select the element with maximum residue norm, stored at position $k+1$ of this list (since the previous k points were already solved in the previous k iterations). With the corresponding point, we evaluate the system and solve for the respective vector at such point, v_{k+1}. Now we repeat the operation (11) with the incremented basis, $V_{k+1} = [V\ v_{k+1}]$, for each of the points given by the sorted list. As we generate the new residues, we keep the maximum norm in $R_{\max}$. At each iteration i of the residue generation (where we computed the new norm corresponding to the point ψ_i) we compare the maximum norm, stored in $R_{\max}$, with the stored norm $\left\|r^k\right\|_{i+1}$ of the next point of the sorted list. If $R_{\max}$ is larger, we can stop the procedure, since from (15) we can state that

$$R_{\max} \geq \left\|r^k\right\|_{i+1} \geq \left\|r^{k+1}\right\|_{i+1}. \tag{16}$$

Therefore, we have found the point for which the norm of the residue is maximum at this iteration k+1. Note that we have not updated all the norms, so for the remaining points i+1, i+2, ... we simply keep the old norms. This procedure will update the norms of the pairs when needed.

A more clear depiction of this step is given by Algorithm 1. With this simple comparisons in step 9, we can achieve significant computation savings: only a small fraction of the norms need to be updated at each iteration.

C. HORUS: Linear Moments Upgraded Sampling

The method, as depicted in the previous sections, only computes the zeroth order moment at each selected point. However, we notice that once we have selected a new suitable point to solve, the main cost comes from solving the system at such point. This solve is usually done by performing an LU factorization of the matrix A_j (evaluated at such point), plus the backsolves with the factors and the matrix B. Let us suppose we have a state space description, with an explicit dependence on the variables, e.g., a Taylor series approximation in the form of (2) in combination with a state space representation in (1). Once we have computed the LU factors for the system matrix, and computed the zero order moment by solving the system, we can obtain the multi-dimensional moments [13] at such expansion point with a few inexpensive extra computations.

We propose to generate the sample plus the first moment for every parameter (including the frequency)

$$\begin{array}{l} LU = \mathbf{LU}(A_j) \\ M_{0,0} = z_j = U^{-1}L^{-1}B \\ M_{1,0} = -U^{-1}L^{-1}C_{0\dots0}M_{0,0} \\ M_{1,1} = -U^{-1}L^{-1}(G_{10\dots0} + s_jC_{10\dots0})M_{0,0} \\ \dots \\ M_{1,P} = -U^{-1}L^{-1}(G_{00\dots1} + s_jC_{00\dots1})M_{0,0} \end{array} \tag{17}$$

where $M_{f,i}$ is the moment of order f with respect to the parameter i (frequency is taken as parameter 0). Note that the zeroth moment corresponds to the nominal values. More refined approaches can generate higher order and cross moments following some of the multi-dimensional moment matching approaches already mentioned (for more details see [13], [14]).

Once all the desired moments are computed, the matrix V is updated with

$$V = \mathbf{RRQR}([V \;\; M_{0,0} \;\; M_{1,0} \;\; M_{1,1} \;\; \ldots M_{1,P}]). \tag{18}$$

Equations (17) and (18) can be integrated, respectively, in steps 6 and 7 of Algorithm 1.

We advocate to generate the first order moments with respect to each parameter, including the frequency, for several reasons: first of all, matching more moments or including cross derivatives will increase the number of vectors in the subspace, with the consequent extra cost in the orthonormalization steps.

To generate first-order moments with respect to each parameter is a straightforward and simple approach, that may help in some scenarios to capture linear information around the sampling point. It can be done with little extra cost, and avoids complicated settings for the methodology (such as determining the number of moments to match w.r.t. each parameter, or when the Taylor series has order higher than first). It could also allow us to reduce the discretization of the space of interest, and thus to have a smaller candidate set, as we are capturing not only information at the point, but also on the nearby region. Systems in which the LU factorization of the matrix is very expensive may benefit from this approach, since by generating more vectors at each point it is likely that we reduce the final number of points to sample, i.e., the number of iterations K in (14). On the other hand, it may affect the optimality of the sample selection (e.g., a single sample may cover the same subspace as several moments).

It is important to realize that by computing only the first moments we are not neglecting higher order or cross term information: the multi-dimensional sampling takes care of the combination of parameters, whereas the linear moments help capture the nearby information. Projection of the complete Taylor series in (2) will keep the same multi-dimensional accuracy.

A different issue is how this step modifies the stopping criteria. We propose to maintain the same criteria, although the norm to check should be the norm of the zero order moment, since that is the one that contains the energetic information of the system at the sampling point. Furthermore, vectors related to higher order moments may suffer from deflation, leading to false information in terms of error convergence.

VI. Simulation Results

In this section, we present the capabilities of the proposed algorithms. ARMS corresponds to the standard method, and HORUS stands for the improvement with first-order moment matching. In both cases, the sorted residues improvement is applied. For comparison, we will use a (variational) Poor Man's TBR (PMTBR) approach [9], [12] based on other sampling schemes. In the single dimension case (frequency) linear or logarithmic sampling will be used. For multi-dimensional (parameterized) cases, the uniform frequency sampling will be combined with pure random sampling in the parameter space. In some tests, a standard TBR [4] and moment matching approaches, namely parameterized interconnect macromodeling algorithm via a two-directional Arnoldi process (PIMTAP) [14], will be used to illustrate certain points.

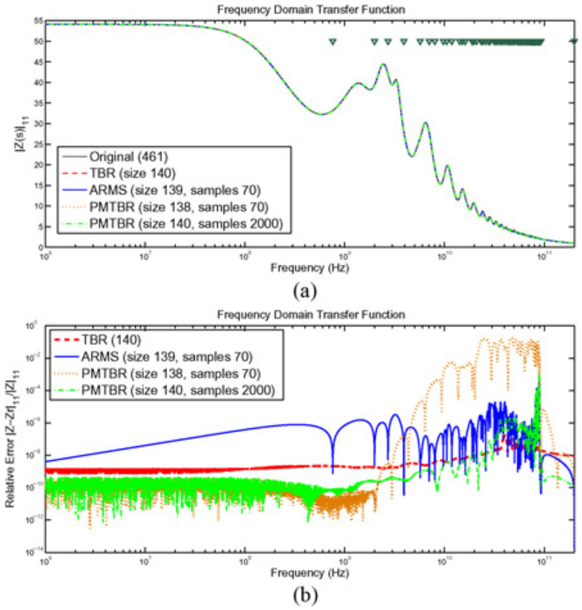

Fig. 1. SISO RLC example. (a) Transfer function for the original and ROMs. Triangles indicate ARMS samples. (b) Relative error for all the ROMs with respect to the original.

The ARMS, HORUS, and PMTBR methods were implemented in C/C++, where the highly efficient SuiteSparse package [22], [26], [27] was used for the solve routines, whereas the rest of the sparse functionalities were self made, and far less efficient.

As benchmarks we will use two examples depending solely on the frequency, and two parameterized systems, that will allow us to validate the underlying theory. RLC is a modified nodal analysis (MNA) formulation of a 2-port system modeled with distributed RLC segments. Although simple, this example is very challenging in terms of sample selection, due to the large frequency range (0–100 GHz) and the flat response at low frequencies and large resonances at high frequencies, that continue with small ripples until the transfer function falls to zero (see Fig. 1 for illustration). PEEC C is a well-known 304-states partial element equivalent circuit (PEEC) example, which has already appeared in the literature [5], and which has very sharp resonances. pRLC is a parameterized MNA formulation of five connected lossy lines. Each line is divided into three parts modeled with 100 RLC segments in which the p.u.l. values of each part depend locally on three parameters, which a total of 45 parameters. The parameters are artificial and modify the R, L, and C values of certain regions up to 20% of their nominal value. The frequency dependent transfer function is the same as in the RLC, but the parameters modify the amplitude and location of the resonances. K-spiral is a RCK second-order EM model of an integrated spiral inductor, whose side length varies up to 37 μm around the nominal value of 187 μm. It is modeled as a second-order model (i.e., the system equations have a $(Y + sC + K/s)x = Bu$ structure), with 5124 resistors, 20 736 capacitors and 381 120 susceptances, and each matrix is approximated with a fourth-order Taylor series on the parameters. The parameter modifies the amplitude and frequency of the transfer function peak.

TABLE I

BENCHMARKS CHARACTERISTICS AND REDUCTION RESULTS

Example	RLC	PEEC C	pRLC	K-Spiral
Model Order	MNA First	PEEC First	MNA First	RCK Second
Size/#Ports NNZ	451/2 1381	304/1 18 982	7551/1 22 531	89 134/1 613 102
#Param./#Terms	0/1	0/1	45/46	1/5
Freq. (GHz)	0–100	0.001–5	0–100	40–60
Validation	10 000	10 000	116 000	6000
PMTBR Size Solves Abs. error Rel. error Time	 349 100 (Log) 9.40e-2 2.4% 2″	 46 23 (Lin) 1.1e-5 4.2% < 1″	 396 200 (Log) 5.04e-1 18.9% 208″	 54 27 (Lin) 7.91e-2 0.002% 1439″
ARMS ROM size Solves/cand. Abs. error Rel. error Time	 113 30/800 6.8e-2 2.1% 27″	 44 22/450 1.5e-6 0.6% 7″	 170 86/1200 1.82e-1 8.8% 3700″	 40 20/700 4.64e-2 0.004% 3323″
HORUS ROM size Solves/cand. Abs. error Rel. error Time	 127 18/800 6.06e-2 1.9% 23″	 47 14/450 4.0e-6 1.5% 4″	 229 10/1200 3.51e-1 10.5% 2895″	 41 12/700 5.71e-2 0.002% 2333″

Table I presents the characteristics of the benchmarks, and the results of the reduction with (variational) PMTBR, ARMS, and HORUS (ROM sizes are defined for a similar accuracy). It can be seen that ARMS and HORUS consistently generate better results with more compressed models due to better sample selection. On the other hand, the cost of the generation is larger due to the extra computations for the sample selection. However, it is important to notice that for most of the examples presented the solve cost is negligible (PEEC example is small, and the distributed RLC and pRLC models have a tri-diagonal matrix structure), whereas the non-optimized sample selection sparse routines take most of the elapsed time. The K-spiral example points out the escalation trend of the algorithm for larger systems, in which the solve time starts to be more dominant.

A. Singular Value Convergence and Optimality

In this subsection, we will show the optimality capabilities for sample selection of the ARMS method.

TBR, PMTBR, and ARMS models are generated for a SISO version of the RLC example. For the ARMS approach, an initial fine candidate set of 2000 points is provided, and a tolerance of 1e-4 is set. The method selects 70 samples to generate a 139-states ROM. For comparison, the size of the TBR model is set to 140. Two PMTBR models are built: one with 70 samples in the domain, and the other with the same 2000 samples used as initial candidate set by ARMS.

Fig. 1 shows the original and ROMs transfer functions, along with the relative error for the ROMs. It can be seen that the PMTBR model with 70 samples exhibits a relatively high error around 100 GHz, due to the small number of samples. The ARMS and 2000-samples PMTBR models have almost the same behavior, with relative errors under 1e-4, although ARMS only solves the system at 70 of the 2000 samples. Note that the frequency points solved by ARMS (indicated by the triangles) are gathered in the region where most of the dynamics take place.

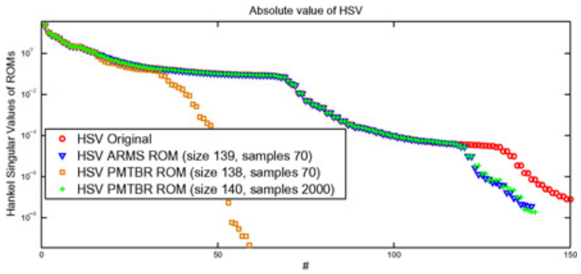

Fig. 2. SISO RLC example. Leading Hankel SV of original, PMTBR, and ARMS models. The PMTBR ROMs are generated with 70 and with 2000 samples, whereas the ARMS ROM is obtained solving 70 samples out of the same 2000 samples as in the PMTBR case.

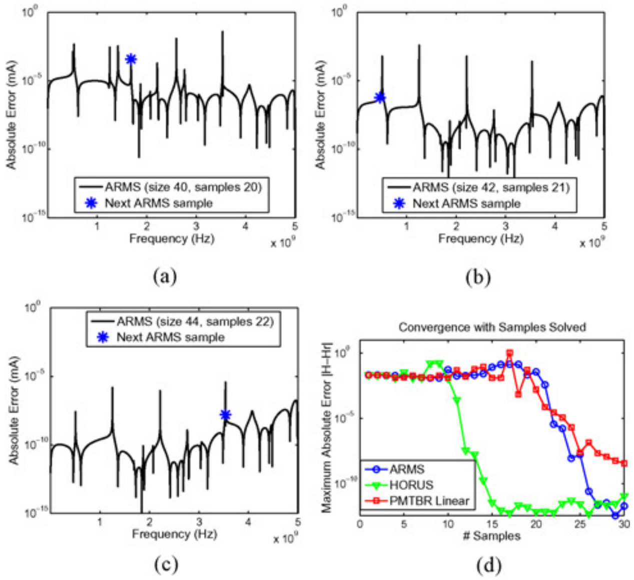

Fig. 3. PEEC C example. (a)–(c) Transfer function error with 20, 21, and 22 samples, and placement of the 21, 22, and 23rd points, respectively. (d) Convergence of the error with the number of samples for PMTBR with linear sampling, ARMS, and HORUS.

Fig. 2 shows the leading Hankel singular values of the original model and the ROMs (i.e., the HSV obtained by solving the Lyapunov equations with the reduced matrices). This gives us an idea of the optimality of the ROMs, since the HSV are related to the energy transference from input to output (see [4] for details). PMTBR with 70 samples is unable to capture the original behavior, whereas with 2000 samples it does a pretty good job. ARMS exhibits almost the same performance as PMTBR with 2000 samples, indicating that it does an excellent job in selecting the most relevant samples to obtain a good ROM (the maximum absolute difference between the HSVs of the original and the ARMS ROM is 1.04e-4). In a sense we could state that the ARMS methodology is able to perform the same job as the SVD, but without the need of solving the 2000 samples (i.e., selecting the good samples before solving the system).

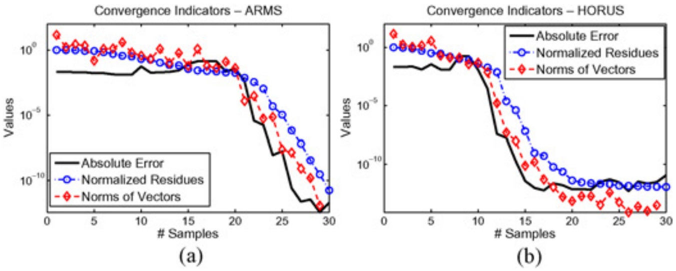

Fig. 4. PEEC C example. Convergence and stopping indicators for (a) ARMS and (b) HORUS.

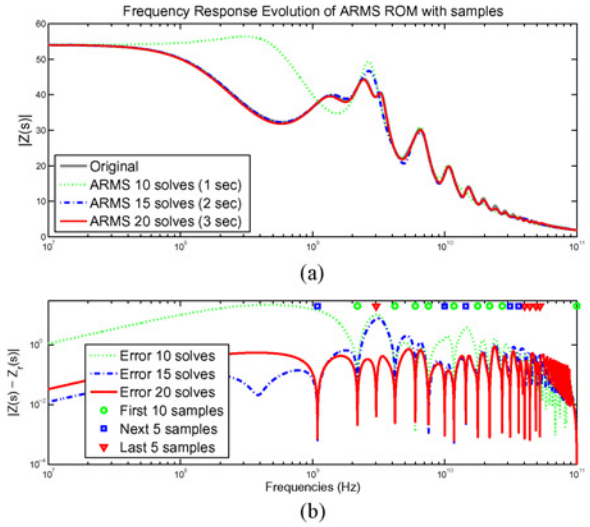

Fig. 5. SISO RLC example. (a) Transfer function for the original and the evolution of the accuracy of ARMS with 10, 15, and 20 samples. (b) Evolution of the absolute error for the same ROMs, and the evolution of sample selection.

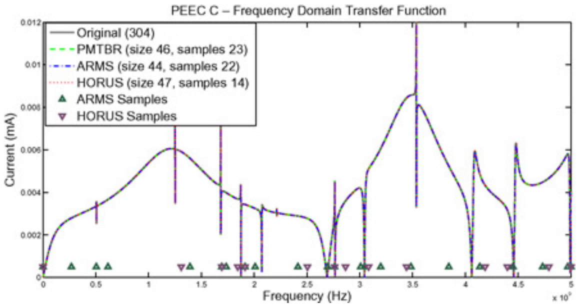

Fig. 6. PEEC C example. Transfer function for ARMS, HORUS, and PMTBR. ARMS and HORUS samples are marked with triangles.

B. Point Selection and Error Convergence

In this section we demonstrate the capabilities of our point selection method, and the fast convergence to a good solution of the proposed methodology.

In Fig. 3, we show a set of consecutive sample points automatically selected by the ARMS algorithm for the PEEC C example (for a transfer function plot see Fig. 6), without any knowledge of the system beyond the minimum and maximum frequencies. Fig. 3 has four plots: the first three present the absolute error of the ARMS ROM with the current number of samples, and marked with a star the frequency where ARMS is going to place the next sampling point. We can see that the samples are placed in regions of large error, near resonances or in regions in which the sample will significantly reduce the overall error. The last plot shows the convergence of the error versus the number of samples for PMTBR, ARMS, and HORUS. The models generated with ARMS exhibit a faster convergence of the error than PMTBR for the same number of samples. HORUS convergence is even faster, needing a smaller number of samples (as it uses two moments per sample). Notice, however, that the sizes of the ARMS and HORUS are similar for equivalent error (i.e., HORUS convergence in terms of ROM size is similar to the one of ARMS).

Fig. 4 shows the same error convergence for ARMS and HORUS for the PEEC C example, along with the information of the stopping criteria. It can be seen that although not perfect indicators of the error, the vector norm and normalized residue provided an automatic approximation for the error convergence.

Fig. 5 shows the quick convergence of ARMS to an acceptable approximation in the SISO RLC example. With 20 samples and in 3″ the approach is able to capture the main resonances, whereas future samples will basically refine the model to capture the high frequency small ripples when the transfer function falls to zero.

Table II presents a similar result for the pRLC example. In this case, ARMS and HORUS results are presented for different numbers of samples and with a different initial candidate set, and benchmarked against linear and logarithmic based PMTBR. It is clear that both ARMS and HORUS are able to quickly converge to good results, whereas a linear sampling is unable to generate good results, and the convergence of the logarithmic based sampling is much slower, generating larger ROMs for equivalent accuracy. Also, the table shows that although the number of initial candidates can have an impact on the accuracy, its effect on the convergence and the generation of accurate models is not critical. On the other hand, the effect of the number of candidates on the elapsed time is quite relevant. This is very encouraging since as previously mentioned, residue computation is embarrassingly parallel. The comparison of CPU times clearly shows that in our non-optimal code, residue computation is the most time consuming task. Therefore large computational improvements may be achieved by parallelization, which is a target of future research. A different feature is that both ARMS and HORUS suffer of an important increase in the computational cost with the number of iterations to perform (i.e., the size of the ROM). Nonetheless, in such cases it is worth spending more time in generating a reliable and more compressed model, which can lead to savings in simulation times.

C. Algorithm Accuracy and Automation

In this section, we present the algorithm features in terms of accuracy and automation, in comparison with PMTBR. It is important to notice that the comparison is not fair (to our disadvantage), since the uniform PMTBR approach needs to know the number of samples beforehand, whereas the ARMS and HORUS approaches do the selection of the number and

TABLE II
pRLC EXAMPLE: ROMs EVOLUTION WITH NUMBER OF SOLVES

PMTBR	Solves (Lin)	25	50	75	100
	ROM size	49	99	149	199
	Time	5″	16″	27″	56″
	Abs. error	23.325	18.210	6.668	2.745
	Solves (Log)	25	50	75	100
	ROM size	49	99	148	198
	Time	5″	16″	27″	56″
	Abs. error	9.154	2.562	1.544	1.098
ARMS	Solves/cand.	25/600	50/600	75/600	100/600
	ROM size	48	98	148	198
	Time	121″	534″	1407″	3499″
	Abs. error	3.460	0.988	0.412	0.125
	Solves/cand.	25/1200	50/1200	75/1200	100/1200
	ROM size	48	98	148	198
	Time	205″	920″	2438″	6740″
	Abs. error	3.880	0.697	0.254	0.107
HORUS	Solves/cand.	3/600	4/600	5/600	6/600
	ROM size	62	107	137	162
	Time	26″	73″	491″	1244″
	Abs. error	28.261	2.564	1.501	0.694
	Solves/cand.	3/1200	4/1200	5/1200	6/1200
	ROM size	67	108	139	159
	Time	58″	176″	1020″	2194″
	Abs. error	17.948	2.041	1.130	0.754

placement of the samples at run time. For ARMS and HORUS, we only fix the minimum and maximum frequencies, and the tolerance, whereas PMTBR we fix the number of samples required for a similar accuracy.

For the PEEC C example, Fig. 6 shows the transfer function for the original and ROMs, and sample placement for ARMS and HORUS. In all cases the accuracy is very good, and the responses of the original and ROM models appear indistinguishable, but the sample points selected by ARMS and HORUS (also shown) are placed in regions of large variation of the transfer function, or near sharp resonances, yielding a more accurate model than the PMTBR approach (the maximum error can be seen in Table I).

Fig. 7 shows the absolute error distribution for the more complex parameterized pRLC benchmark. Results are shown for the variational PMTBR, for the multi-dimensional moment matching PIMTAP method, and for the proposed ARMS and HORUS algorithms. Notice that the x-axis scale is the same. All the ROMs have a similar size, and we have included a moment-matching approach for comparison. Results are similar to previous cases. The moment matching approach is unable to provide good accuracy, and the PMTBR model shows a large deviation, which indicates that the error is small in some regions (oversampling) whereas it is very large in others (undersampling). The HORUS model needs a much smaller number of samples than ARMS to generate model with acceptable accuracy, which is the main benefit of generating first-order moments in the upgraded sampling. On the other hand, the ARMS model is more optimal, in the sense that better accuracy is obtained with a smaller model. Therefore, when ROM size is critical, ARMS would be the method of choice. On the other hand, cases in which a good (near optimal) basis needs to be quickly generated, HORUS would be a better option.

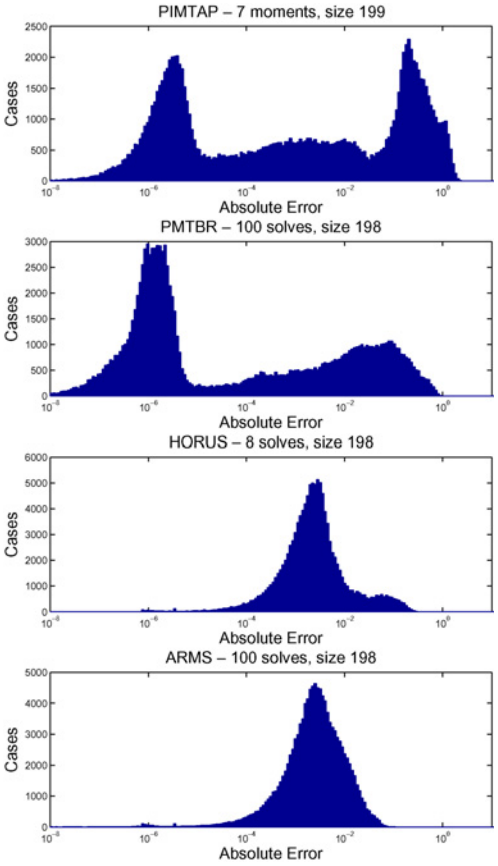

Fig. 7. pRLC: simulation on 116 000 multi-dimensional points. On vertical axis are the number of occurrences, whereas on horizontal axis are the maximum absolute error of the ROM with respect to the original parameterized model. ROMs are, from top to bottom, PIMTAP, PMTBR, HORUS, and ARMS, all with similar ROM size.

VII. Conclusion

An efficient algorithm for automatic selection and minimization of the number of samples to use in a MIMO multi-point MOR framework has been presented, which:

1) uses global information of an arbitrary set of samples avoiding solving the system for all of them; in fact, the method minimizes the number of solves to perform;
2) selects the most suitable point to solve for automatically, by monitoring the residue (used as a proxy for the error) at each candidate sample, with the objective of maximizing the projection subspace;
3) is general and independent of the representation, and has been successfully applied to single and multiple dimensions, with potential application to non-linear models.

The procedure was integrated into a set of MOR algorithms, offering a reliable, robust, and accurate framework able to deal with different cases. This framework provides multiple advantages, namely the following.

1) *Automation*. The method automatically selects the best samples from a initial candidate set, without any prior system information, and automatically determines when the model is accurately captured, via cheap and easy to monitor criteria. Previous system information can be

easily incorporated with potential efficiency improvement.

2) *Efficiency.* Although it requires some extra computations, the algorithm uses the information of a large set of samples by solving the system for a minimum number of points. It has been demonstrated that it can efficiently handle large systems.
3) *Optimality*. Although the method does not provide optimal overall models (in the TBR sense), it can provide local models (in the defined multi-dimensional region) close to optimal. However, there is a tradeoff between efficiency and minimum reduced order.
4) *Generality*. It can be applied to systems of different nature (MIMO, linear, parameterized, and so on) and in different representations (first, second order, Taylor series, and so on) with minimum modifications.
5) *Reliability*. The framework is very robust due to the underlying multi-point and projection. In addition, it ensures a high level of confidence in the accuracy of the ROM for the region covered by the initial candidate set.

Results demonstrate that it outperforms other sampling schemes and reduction methodologies in both single and multiple dimensions, for a variety of benchmarks.

Acknowledgment

The authors would like to thank G. Ciuprina and D. Ioan (LMN, Politehnica University of Bucharest, Bucharest, Romania), K.-J. van der Kolk and N. van der Meijs (TU Delft, Delft, The Netherlands), and R. Janssen and W. Schilders (NXP Semiconductors Research, The Netherlands). They would also like to specially thank M. E. Hochstenbach (TU Eindhoven, Eindhoven, The Netherlands) for pointing them into the low-discrepancy sequences.

References

[1] J. F. Villena and L. M. Silveira, "ARMS: Automatic residue-minimization based sampling for multi-point modeling techniques," in *Proc. DAC*, Jul. 2009, pp. 951–956.

[2] J. F. Villena and L. M. Silveira, "HORUS: High-dimensional model order reduction via low moment-matching upgraded sampling," in *Proc. Des. Autom. Test Eur. Conf. Exhibit.*, Mar. 2010, pp. 465–470.

[3] W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, Eds., *Model Order Reduction: Theory, Research Aspects and Applications* (Mathematics in Industry, vol. 13: The European Consortium for Mathematics in Industry). Berlin, Germany: Springer, 2008.

[4] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. Philadelphia, PA: SIAM, 2005.

[5] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 14, no. 5, pp. 639–649, May 1995.

[6] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Des.*, vol. 17, no. 8, pp. 645–654, Aug. 1998.

[7] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Automat. Control*, vol. 26, no. 1, pp. 17–32, Feb. 1981.

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

[9] J. R. Phillips and L. M. Silveira, "Poor man's TBR: A simple model reduction scheme," *IEEE Trans. Comput.-Aided Des.*, vol. 24, no. 1, pp. 43–55, Jan. 2005.

[10] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *Am. Inst. Aeronautisc Astronautics J.*, vol. 40, no. 11, pp. 2323–2330, Nov. 2002.

[11] J. R. Singler, "Approximate low rank solutions of Lyapunov equations via proper orthogonal decomposition," in *Proc. Am. Control Conf.*, Jun. 2008, pp. 267–272.

[12] J. Phillips, "Variational interconnect analysis via PMTBR," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2004, pp. 872–879.

[13] L. Daniel, O. C. Siong, S. C. Low, K. H. Lee, and J. K. White, "A multiparameter moment-matching model-reduction approach for generating geometrically parametrized interconnect performance models," *IEEE Trans. Comput.-Aided Des.*, vol. 23, no. 5, pp. 678–693, May 2004.

[14] Y. Li, Z. Bai, Y. Su, and X. Zeng, "Model order reduction of parameterized interconnect networks via a two-directional Arnoldi process," *IEEE Trans. Comput.-Aided Des.*, vol. 27, no. 9, pp. 1571–1582, Sep. 2008.

[15] Z. Zhu and J. Phillips, "Random sampling of moment graph: A stochastic Krylov-reduction algorithm," in *Proc. Des. Autom. Test Eur. Conf. Exhibit.*, Apr. 2007, pp. 1502–1507.

[16] B. N. Bond and L. Daniel, "A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems," *IEEE Trans. Comput.-Aided Des.*, vol. 26, no. 12, pp. 2116–2129, Dec. 2007.

[17] Y. Bi, K.-J. van der Kolk, and N. van der Meijs, "Sensitivity computation using domain-decomposition for boundary element method based capacitance extractors," in *Proc. IEEE CICC*, Sep. 2009, pp. 423–426.

[18] T. El-Moselhy, I. Elfadel, and D. Widiger, "Efficient algorithm for the computation of on-chip capacitance sensitivities with respect to a large set of parameters," in *Proc. ACM/IEEE DAC*, Jun. 2008, pp. 906–911.

[19] D. Ioan, G. Ciuprina, and W. H. A. Schilders, "Parametric models based on the adjoint field technique for RF passive integrated components," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1658–1661, Jun. 2008.

[20] S. Gugercin, A. Antoulas, and C. Beattie, "A rational Krylov iteration for optimal H_2 model reduction," in *Proc. 17th Int. Symp. Math. Theory Netw. Syst.*, Jul. 2006, pp. 1665–1667.

[21] L. M. Silveira and J. R. Phillips, "Resampling plans for sample point selection in multipoint model order reduction," *IEEE Trans. Comput.-Aided Des.*, vol. 25, no. 12, pp. 2775–2783, Dec. 2006.

[22] T. A. Davis, *Direct Methods for Sparse Linear Systems* (The Fundamentals of Algorithms). Philadelphia, PA: SIAM, Sep. 2006.

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

[24] C. H. Bischof and G. Quintana-Ortí, "Computing rank-revealing QR factorizations of dense matrices," *ACM Trans. Math. Softw.*, vol. 24, no. 2, pp. 226–253, 1998.

[25] P. Bratley, B. L. Fox, and H. Niederreiter, "Implementation and tests of low-discrepancy sequences," *ACM Trans. Modeling Comput. Simul.*, vol. 2, no. 3, pp. 195–213, Jul. 1992.

[26] T. A. Davis, I. S. Duff, P. Amestoy, J. R. Gilbert, S. Larimore, E. Palamadai Natarajan, Y. Chen, W. W. Hager, and S. Rajamanickam. *Suitesparse: A Suite of Sparse Matrix Packages* [Online]. Available: http://www.cise.ufl.edu/research/sparse/SuiteSparse

[27] T. A. Davis, "Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.*, vol 30, no. 2, pp. 196–199, Jun. 2004.

Jorge Fernández Villena (M'06) was born in Avilés, Spain. He received the Engineers degree in telecommunications from Cantabria University, Cantabria, Spain, in 2005, and the Ph.D. degree in electrical and computer engineering from Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, in 2010.

He is currently a Researcher with the Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa (INESC ID), Lisbon. His current research interests include IC interconnect modeling and simulation, with emphasis on parameterized model order reduction.

Luís Miguel Silveira (S'85–M'95–SM'00), photograph and biography not available at the time of publication.