

Scalable Models of Microwave System Responses using Sequential Sampling on Unstructured Grids

Krishnan Chemmangat*, Francesco Ferranti*, Tom Dhaene* and Luc Knockaert*

Department of Information Technology, Ghent University - IBBT, Gaston Crommenlaan 8 Bus 201, B-9050, Gent, Belgium.

SUMMARY

This paper presents two sequential sampling algorithms for the macromodeling of parameterized system responses in model-dependent sampling frameworks. The construction of efficient algorithms for the automatic selection of samples for building scalable macromodels of frequency-domain responses is addressed in this paper. The sequential sampling algorithms proposed here are tailored towards the application of local scalable macromodeling schemes on unstructured design space grids. Two pertinent examples are considered. For the first one, different algorithms are applied and a comparison is made in terms of the number of samples generated, accuracy and CPU time. As a second example, four design variables are taken into account with one of the proposed algorithms and the generated model is used in a frequency-domain optimization.

Received . . .

KEY WORDS: Interpolation, Macromodels, Scalable macromodeling, Sequential sampling.

1. INTRODUCTION

Design space exploration, design optimization and sensitivity analysis of electromagnetic (EM) systems often require expensive simulations using EM solvers which normally provide high accuracy at a significant cost in terms of memory storage and computing time. The computational complexity of these EM solvers often results in a design cycle that is costly in terms of execution time, which might not be acceptable in practice.

An alternative is to develop accurate and efficient scalable macromodels which approximate the complex behavior of EM systems, characterized by frequency and additional design parameters, such as geometrical or substrate features. Scalable macromodeling of EM systems has attracted a lot of attention during recent years [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. However, one of the key challenges in these modeling approaches, which requires further research, is the optimal selection of data samples over the design parameter space, in order to limit the total number of expensive EM simulations [1, 7, 2, 4, 3, 5, 14].

Sequential sampling techniques can be classified into three main categories, i.e., the input-based methods, the output-based methods and the model-based methods. The sequential Design of Experiments (DoE) method falls into the *input-based* category, where the input design space is filled

*Correspondence to: Krishnan Chemmangat, Department of Information Technology, Ghent University - IBBT, Gaston Crommenlaan 8 Bus 201, B-9050, Gent, Belgium. E-mail: krishnan.cmc@intec.ugent.be

Contract/grant sponsor: This work was supported by the Research Foundation Flanders (FWO) and by the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office.

according to some measure depending on the density of the selected samples [16, 17]. The *output-based* sampling schemes depend on the output system response in order to sequentially select new samples in the design space [14, 18]. A recent work on sequential sampling of scattering parameter responses (S-responses) uses output S-responses along with an exploration-exploitation-based approach [14]. In the exploration phase, which searches the design space for unidentified regions, a space filling technique is used to fill the design space uniformly, whereas in the exploitation phase, which identifies potentially interesting and dynamic regions of the design space, a local estimate of the gradient is used to select new samples. In contrast to the other two categories, *model-based* sampling schemes depend on intermediate macromodels to select the distribution of new samples [1, 7, 2, 4, 3, 5].

Apart from the above mentioned categories, there are global and local sequential sampling methods. *Global* sequential sampling algorithms build a single model for the complete design space. Multi-dimensional Cauchy methods are one of the popular global sequential sampling schemes [1, 2, 3, 6]. In [1], two different sampling approaches are mentioned. The first approach which uses a stable recursive Burlisch-Stoer algorithm has an inherent limitation since only the last parameter can be sampled and the rest of the parameters must lie on a fully filled grid. The second method, which uses a multi-dimensional rational function expansion can be very ill-conditioned at higher dimensions, limiting the applicability of this method to relatively simple problems. Total Least Square (TLS) algorithm was used in [2] to solve for the coefficients of the multivariate rational model with a QR factorization, and for each added samples a QR update is used instead of full QR decomposition to gain computational time. An adaptive multivariate rational fitting is reported in [3] which uses Tchebishev orthogonal polynomials to improve the conditioning of the matrices to be solved. In [6], a multi-dimensional rational approximation is built using convex optimization with linear constraints. These constraints ensure that the algorithm is stable and the value of the interpolated function lies within the physical bounds. Neural network and radial basis function-based sequential sampling method are also found in the literature [4, 5] which are also global modeling schemes. However, such global modeling schemes [1, 2, 3, 4, 5, 6] suffer from the following limitations:

1. These methods are unable to preserve physical properties such as stability and passivity of the generated macromodels over the design space of interest which are very important while performing time-domain simulations [19].
2. For most of these methods, the initial sampling of the design space is very important. If the initial sampling is not adequate, the accuracy of the initial models which are used for sequential sampling may be very inaccurate leading to divergence.
3. For relatively high dimensions the memory requirement can be prohibitively high limiting their applicability.

This paper describes two sequential sampling algorithms for selecting the optimum number of samples such that accurate scalable models for parameterized system responses can be generated. The methods presented here are quite different from the approach in [1, 2, 3, 6, 4, 5, 14], since the proposed algorithms are *local* and work on local n-box regions of the design space. This creates the possibility of a tree-based implementation of the algorithms similar to [20, 21], reducing the model evaluation time and making them portable to parallel computing platforms. The algorithms based on exploitation of the design space result in a design space which is suitable for the application of different passivity-preserving local scalable macromodeling algorithms [9, 10, 11, 12, 13, 15]. Also, in contrast to the global sequential sampling methods [1, 2, 3, 6, 4, 5], once the passivity is enforced on the sampled points, the local interpolation ensures that the models are stable and passive over the complete design space of interest and hence can be used in time-domain simulations. Also, the two sequential sampling algorithms described here are applied on model-based frameworks.

The paper is organized as follows. Section 2 presents some important aspects of sampling schemes. Section 3 describes the two proposed sampling algorithms in detail. Different pertinent numerical examples are presented in order to validate and compare the different schemes discussed in the paper, and the corresponding results are presented in Section 4.

2. SEQUENTIAL SAMPLING

A sequential sampling algorithm selects those samples which allow to build an accurate model with respect to some error measures. A design space consists of all the design variables such as layout variables or substrate features. Since frequency is a special parameter whose behavior can be accurately modeled using a rational function, it is sampled separately and is not considered as a part of the design space. If needed, an Adaptive Frequency Sampling (AFS) can be used to sample the frequency axis. In this paper, the word *design sample point* denotes the frequency-domain response of a microwave system for a particular design configuration. The sampled system response data obtained through an EM solver are used to build macromodels which accurately describe the parameterized input-output behavior of the original complex system with a predefined level of accuracy. Some important aspects of a suitable sequential sampling strategy are briefly explained in subsections 2.1 and 2.2.

2.1. Error measure

Another important aspect of a suitable sequential sampling algorithm is to define proper error criteria for choosing new samples. During the evolution of the design space any new point added in the design space considerably reduces a judiciously chosen error measure between the system response of the original simulation model and the intermediate macromodel. Several error measures can be used to compare frequency-domain responses. The maximum absolute error between the original frequency response $H_{i,j}$ and the macromodel $R_{i,j}$ can be used,

$$E^{\text{MaxAbs}}(\vec{g}) = \max_{i,j,k} \left(\left| R_{i,j}(s_k, \vec{g}) - H_{i,j}(s_k, \vec{g}) \right| \right) \quad (1)$$

$$i = 1, \dots, P_{in}, j = 1, \dots, P_{out}, k = 1, \dots, N_s$$

with number of input ports P_{in} , output ports P_{out} and frequency samples N_s (s_k is the complex frequency or Laplace variable). However, the maximum absolute error can be misleading when comparing two frequency responses since this can give extreme values, e.g. at the resonance peaks. An alternative is to use mean error measures like the mean absolute error (MAE):

$$E^{\text{MAE}}(\vec{g}) = \frac{\sum_{i=1}^{P_{in}} \sum_{j=1}^{P_{out}} \sum_{k=1}^{N_s} |R_{i,j}(s_k, \vec{g}) - H_{i,j}(s_k, \vec{g})|}{P_{in} P_{out} N_s}. \quad (2)$$

This paper uses the mean of absolute error (2) for the sequential sampling algorithms, as suggested in [22].

In addition to the absolute error measures discussed above, weighted or relative error measures such as the relative mean error:

$$E_{\text{Rel}}^{\text{MAE}}(\vec{g}) = \frac{\sum_{i=1}^{P_{in}} \sum_{j=1}^{P_{out}} \sum_{k=1}^{N_s} \frac{|R_{i,j}(s_k, \vec{g}) - H_{i,j}(s_k, \vec{g})|}{|H_{i,j}(s_k, \vec{g})|}}{P_{in} P_{out} N_s} \quad (3)$$

can be used while modeling impedance or admittance parameters, as these parameters are not bounded so that good accuracy is obtained over the complete dynamic range.

2.2. Selection of scalable macromodeling method

There are several choices available for a macromodeling scheme to be used in the sequential sampling and the final design space will depend on this selection. As discussed in the introduction, the sequential sampling algorithms considered in this work are local and hence any macromodeling scheme suitable to this scenario could be used.

One choice could be to use multivariate local interpolation methods, e.g., the piecewise multilinear and multivariate simplicial methods [23] on the output frequency response data and build a scalable macromodel. This choice makes the sequential sampling depend on the output frequency response data and the used interpolation scheme.

Another alternative is to use one of the local scalable macromodeling schemes which use the Vector Fitting (VF) technique [24, 25, 26] to build frequency-dependent rational models called *root macromodels* at the selected design space samples and then parameterize them, see [9, 10, 11, 12, 13, 15]. The scalable macromodeling process starts with a set of multivariate data samples $\{(s, \vec{g})_k, \mathbf{H}(s, \vec{g})_k\}_{k=1}^{K_{tot}}$ which depends on frequency and additional design variables. From these data samples, a set of *root macromodels* in pole-residue form are built for a set of design space samples \vec{g}_k by means of VF yielding a set of *root macromodels* $\mathbf{R}(s, \vec{g}_k)$. A pole-flipping scheme is used to enforce strict stability [24] and passivity assessment and enforcement is achieved using robust standard techniques [27, 28] resulting in a set of stable and passive *root macromodels*. The next step of these scalable macromodeling algorithms is the parameterization of the set of *root macromodels* $\mathbf{R}(s, \vec{g}_k)$. In [10, 11], a scalable macromodel is built by interpolating a set of *root macromodels* at an input-output level, while in [9, 12, 13], both poles and residues are parameterized by interpolating the internal state-space matrices, resulting in higher modeling capability with respect to [10, 11]. In [15], a novel enhanced interpolation of *root macromodels* at an input-output level is described, which is based on the use of some coefficients: one coefficient as a multiplicative factor at the input/output level of the system and the other coefficient as a compression or expansion term for the Laplace variable s . It results in high modeling capability and robustness.

Using stability and passivity enforced VF-based scalable macromodeling schemes have the following advantages:

1. They generate rational models which are stable and passive over the complete design space, and therefore suitable for time-domain simulations, which cannot be achieved using the interpolation of raw frequency response data.
2. By selecting a powerful scalable macromodeling scheme, the number of EM simulations can be reduced considerably as shown by the numerical results in Section 4.

3. SEQUENTIAL SAMPLING ALGORITHMS

This section describes two sequential sampling algorithms developed in this work. The proposed sequential sampling algorithms preserve the rectangular nature of the sampling grid such that the existing local scalable macromodeling methods [10, 11, 15] can be directly applied. We call a n -box region of the design space as a subspace or a node.

The following aspects of the proposed sequential sampling methods distinguishes them from the previous methods [1, 2, 3, 6, 4, 5]:

1. The sequential sampling described here preserve the physical properties such as stability and passivity for the entire design space of interest, which is fundamental for time-domain simulations.
2. The methods presented here are local and hence can efficiently be implemented on parallel computing platforms in a tree-based structure.
3. In contrast to the multivariate rational fitting, several frequency-dependent models are identified and then parameterized with respect to design variables. This considerably reduces the utilization of memory resources in high dimensional design spaces.

3.1. Algorithm I: Division at the center of a subspace

Here we assume that the model is prone to a high error value at the center of a subspace. Hence, the idea is to subdivide a subspace at its center, when the error measure is greater than a predefined threshold. For example, let us consider a rectangular subspace for a two dimensional design space with four samples as shown in Fig. 1-a for which a bivariate macromodel is generated. A test sample is selected at the geometric center of the subspace (shown by a gray circle in Fig. 1-b) and the frequency responses between the original EM solver and macromodel are compared at this test point. If the modeling error (2) is found to be greater than a predefined threshold Δ , the subspace is

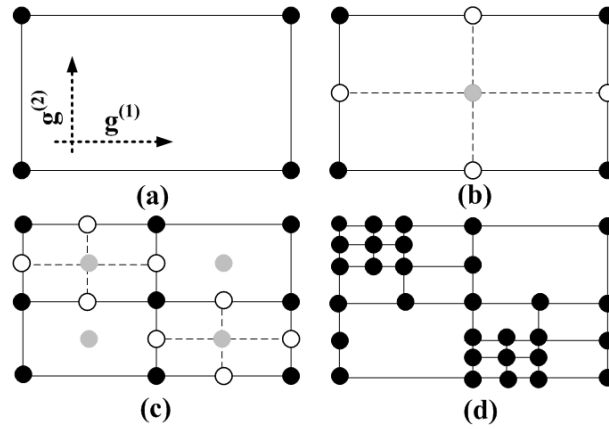


Figure 1. *Algorithm I*: Division of the design space.

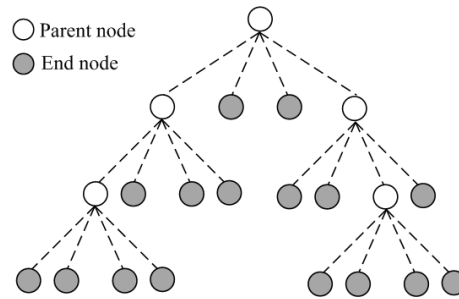


Figure 2. Fig. 1-d represented in a tree structure.

subdivided, generating four new rectangular subspaces. This procedure is repeated until the error at the geometrical center of each subspace is less than the predefined threshold (Fig. 1-c) and the final design space can for example look like Fig. 1-d.

Fig. 2 shows the design space of Fig. 1-d as a tree structure with each node (circle) representing a particular subspace. The branches (dotted lines) represent the connection between a node and its subregions or *child nodes*. Note that each node is divided into four child nodes or in general into 2^N nodes for an N -dimensional subspace. The terminal nodes are represented in gray circles where the required accuracy is achieved. With a tree-based implementation, the advantage is the portability to parallel computing platforms with a reduction of computing time.

The sequential sampling process of *Algorithm I* consists of the following steps:

- I) Generating an initial n -box design space by defining 2^N corner points that define the convex hull of the design space, where N is the number of design variables $\vec{g} = (g^{(1)}, \dots, g^{(N)})$. The number of subspaces $Q = 1$.
- II) Building a macromodel $\mathbf{R}(s, \vec{g})$ for the entire design space with Q subspaces.
- III) For each particular subspace $q = 1, \dots, Q$, checking the error criteria at the center of the subspace.
 - i. IF($Err_q > \Delta$):
 - i. Divide the subspace q into 2^N new subspaces and increment $Q = Q + 2^N - 1$.
 - ii. Increment $q = q + 1$ and go to Step II.
 - ii. ELSE: increment $q = q + 1$.
 - i. IF ($q \leq Q$): Not all subspaces are checked for the error criteria. Go to Step II.
 - ii. ELSE: Go to Step IV.
- IV) Terminating the sequential sampling algorithm.

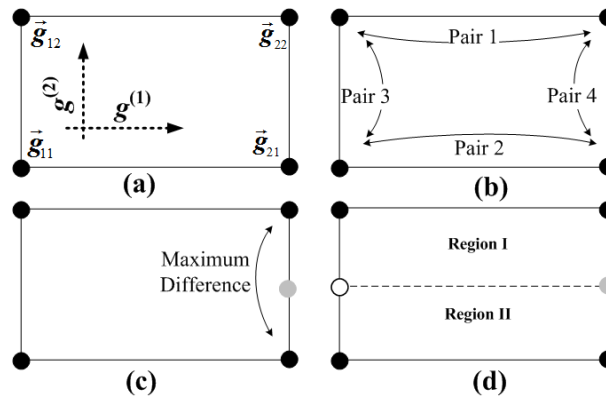


Figure 3. *Algorithm II*: Division of the design space.

3.2. *Algorithm II*: Division at the edge of a subspace followed by final refinement using *Algorithm I*

As in *Algorithm I*, the rectangular grid structure is preserved here. However, in contrast with *Algorithm I*, a subspace is subdivided into two equal halves along the maximum sensitive design variable instead of the geometric center. The maximum sensitive variable in a particular subspace is found in the following way. Consider a bivariate case with parameter vector $\vec{g} = (g^{(1)}, g^{(2)})$ as shown in Fig. 3-a, where the four initial samples are marked by $\vec{g}_{ij} = (g_i^{(1)}, g_j^{(2)})$; $i, j = 1, 2$. Now these points are paired for all four combinations as in Fig. 3-b, and the difference between the macromodel responses $\mathbf{R}(s, \vec{g}_{ij})$ are measured for each pair using the error measure given by (2). Let us assume that the difference between the pairs $\mathbf{R}(s, \vec{g}_{21})$ and $\mathbf{R}(s, \vec{g}_{22})$ is the biggest as shown in Fig. 3-c, then a new point is taken at the middle of the range $[\vec{g}_{21}, \vec{g}_{22}]$, shown by the gray circle, and this is used as a test point to check the accuracy of the macromodel with respect to the original EM solver. If the error is found to be greater than a predefined threshold Δ , the subspace is divided into two *child nodes* along this design variable which happens to be $g^{(2)}$ in Fig. 3.d, by taking additional points (white circles). Finally, the generated design space is further refined using *Algorithm I*.

The sequential sampling process of *Algorithm II* consists of the following steps:

- I) Same as Step I of *Algorithm I*.
- II) Same as Step II of *Algorithm I*.
- III) For each particular subspace $q = 1, \dots, Q$, finding the maximum sensitive design variable and hence the test point as previously explained.
- IV) Checking the error criteria at the test point.
 - i. IF ($Err_q > \Delta$):
 - i. Divide the subspace q into 2 new subspaces and increment $Q = Q + 1$.
 - ii. Increment $q = q + 1$ and go to Step II.
 - ii. ELSE: increment $q = q + 1$.
 - i. IF ($q \leq Q$): Not all subspaces are checked for the error criteria. Go to Step II.
 - ii. ELSE: Go to Step V.
- V) Performing final refinement using *Algorithm I*.
- VI) Terminating the sequential sampling algorithm.

The main advantage of *Algorithm II* as compared to *Algorithm I* is that the number of points added for each subspace division is significantly reduced. Also, if some of the parameters are highly influential, the algorithm divides along that direction, ensuring fine refinement along the highly dynamic parameters instead of dividing all the parameters at the same time as in *Algorithm I*. Please note that, the capability to implement the algorithm as a tree structure is preserved.

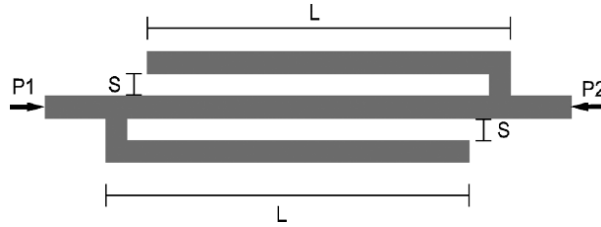


Figure 4. Layout of the DFS band-stop filter.

4. NUMERICAL SIMULATIONS

The proposed algorithms have been implemented in Matlab R2010a[†] and used to drive the ADS Momentum[‡] simulations to generate S-responses at selected samples. The numerical simulations have been performed on a Windows 7 platform on Intel(R) Core(TM)2 Duo P8700 2.53 GHz machine with 2 GB RAM. The proposed sequential sampling algorithms (Section 3) are compared and the effect of the two macromodeling schemes on the sampling is studied on pertinent numerical examples.

Double folded stub microwave filter

In this example, a Double Folded Stub (DFS) band-stop microwave filter on a substrate with relative permittivity $\epsilon_r = 9.9$ and a thickness of 0.127 mm is modeled. The layout of this DFS filter is shown in Fig. 4. The spacing S between the stubs and the length L of the stubs are chosen as design variables in addition to frequency whose ranges are $S \in [0.15, 0.25] \text{ mm}$, $L \in [1.0, 2.5] \text{ mm}$ and frequency $\in [5, 20] \text{ GHz}$.

The S-response matrix $\mathbf{S}(s, S, L)$ has been computed using the ADS Momentum solver and the number of frequency samples has been chosen equal to 31. Fig. 5 shows the parametric behavior of the magnitude of S_{11} as a function of S and frequency for $L = 1.75 \text{ mm}$. Similarly, Fig. 6 shows the magnitude of S_{21} as a function of L and frequency for $S = 0.20 \text{ mm}$. Figs. 7 and 8 show the surface plots for the Figs. 5 and 6 respectively.

The two sequential sampling algorithms described in Section 3 are implemented using two different macromodeling schemes. In the first implementation, a local multilinear interpolation of the raw frequency response data is used to build the intermediate macromodels. The second implementation uses the scalable macromodeling technique described in [15] which perfectly fits into our proposed sequential sampling algorithms. The sequential sampling algorithms are applied on each of these cases, as seen in Table I. Table I compares all proposed schemes in terms of the total number of samples, the worst case mean absolute error over the design space, the CPU time needed to run all the ADS momentum simulations and the CPU time needed for the sequential sampling algorithms. The different sequential sampling experiments performed on the DFS structure are summarized in Table I. The overall target accuracy was set to $\Delta = -50 \text{ dB}$. The number of poles of the *root macromodels* were calculated using an error-based bottom-up approach. The initial number of poles for this bottom-up approach is selected as the smallest order of the *root macromodels* at the corners of the n-box design space. The accuracy of the final sampling grid is assessed using validation samples generated at the midpoint of each subspace of the final grid where the response from the EM solver is compared with the macromodel using the mean absolute error (2). Table I the results of the sequential sampling algorithms with some important parameters.

The following observations can be made from the tabulated results:

- 1) *Algorithm II* generates less number of points in comparison with *Algorithm I*. This is in accordance with the expectations of Section 3.

[†]The Mathworks, Inc., Natick, MA, USA

[‡]Momentum EEs of EDA, Agilent Technologies, Santa Rosa, CA.

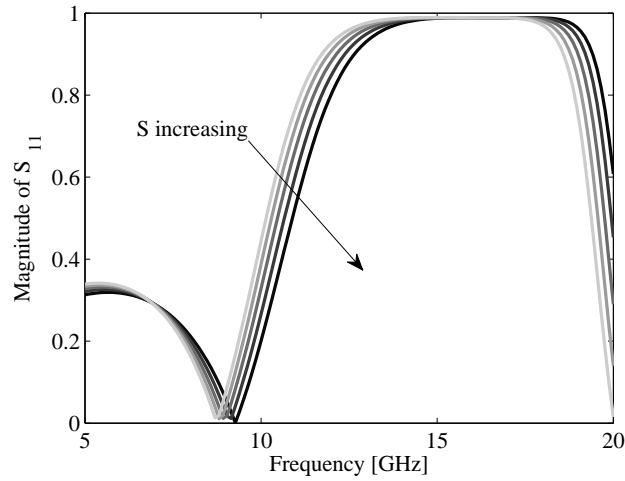


Figure 5. DFS Filter: Magnitude of S_{11} for $L = 1.75$ mm with scalable macromodel generated using *Algorithm II*.

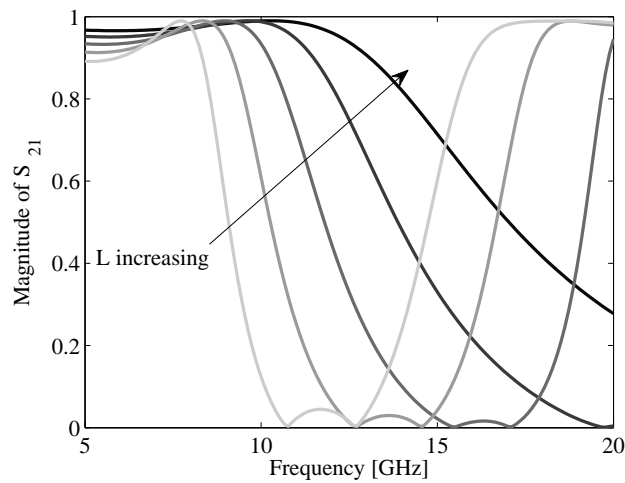


Figure 6. DFS Filter: Magnitude of S_{21} for $S = 0.20$ mm with scalable macromodel generated using *Algorithm II*.

Table I. DFS Filter: Comparison of different sampling strategies.

Sampling method		Output data interpolation		Scalable macro-modeling [15]	
Sampling algorithm		I	II	I	II
Number of samples		1460	286	81	51
Accuracy (MAE) [dB]		-50.00	-50.01	-52.01	-50.25
CPU time [s]	Data generation	171363	56962	15908	10968
	Alg. execution	372	35	502	334

- 2) Using the scalable macromodeling technique of [15], the number of samples generated is very low in comparison with the approach where the raw frequency response data is interpolated as shown in Fig. 9. This indicates the high modeling capability of the scalable macromodeling method [15] and its effect on the sampling.

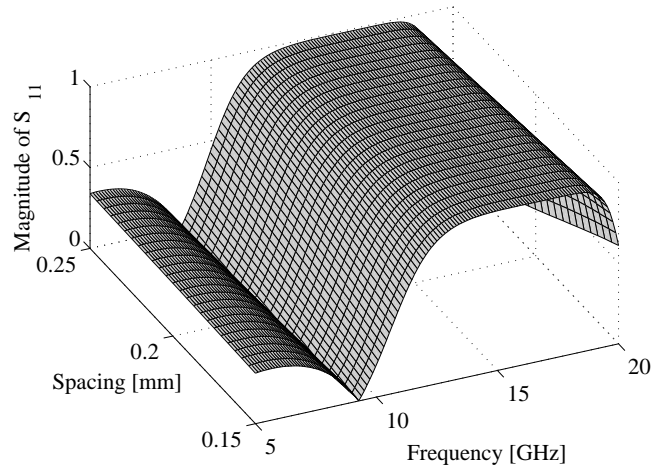


Figure 7. DFS Filter: Magnitude of S_{11} for $L = 1.75$ mm with scalable macromodel generated using *Algorithm II*.

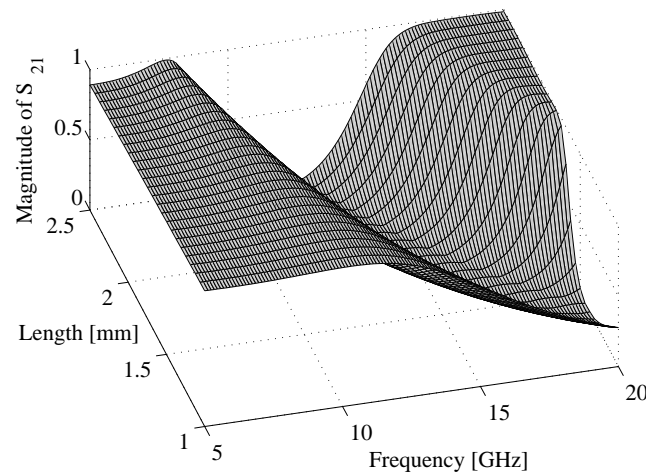


Figure 8. DFS Filter: Magnitude of S_{21} for $S = 0.20$ mm with scalable macromodel generated using *Algorithm II*.

As a rule of thumb, a good choice is to use the *Algorithm II* (Section 3.2) to sample the design space such that the influence of each design variable is taken into consideration and a good accuracy is achieved. The scalable macromodeling scheme described in [15] can be used to achieve a considerable gain in terms of the number of required EM simulations.

Hairpin bandpass microwave filter

A microwave hairpin bandpass filter on a substrate with relative permittivity $\epsilon_r = 9.9$ and a thickness of 0.635 mm is modeled in this example. The layout of this filter is shown in Fig. 10. Two spacings S_1 and S_2 and two lengths L_1 and L_2 are chosen as design variables (see Fig. 10) in addition to frequency whose ranges are $S_1 \in [0.25, 0.35]$ mm, $S_2 \in [0.65, 0.75]$ mm, $L_1 \in [9.5, 14.5]$ mm, $L_2 \in [2.75, 3.25]$ mm and frequency $\in [1.5, 3.5]$ GHz.

Fig. 11 shows the parametric behavior of the magnitude of S_{11} as a function of S_1 and frequency, other values being kept at the mean value of the design space. Similarly, Fig. 12 shows the magnitude

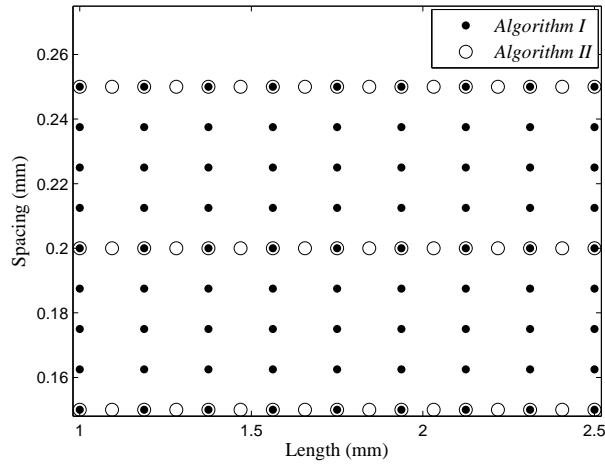


Figure 9. DFS Filter: Design space for *Algorithm I* and *Algorithm II* using scalable macromodeling method [15].

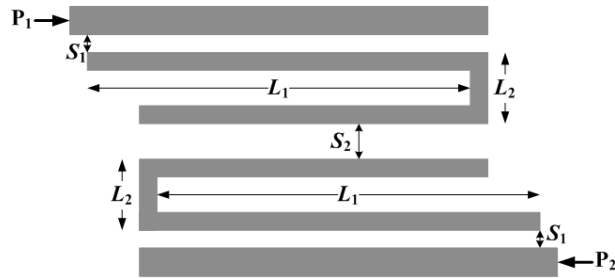


Figure 10. Layout of the microwave hairpin bandpass filter.

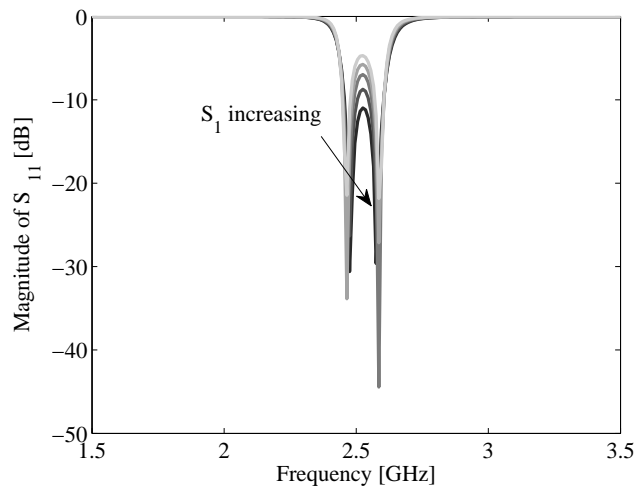


Figure 11. Hairpin Filter: Magnitude of S_{11} as a function of S_1 with scalable macromodel generated using *Algorithm II*.

of S_{21} as a function of L_1 and frequency. Figs. 13 and 14 show the surface plots for the Figs. 11 and 12 respectively.

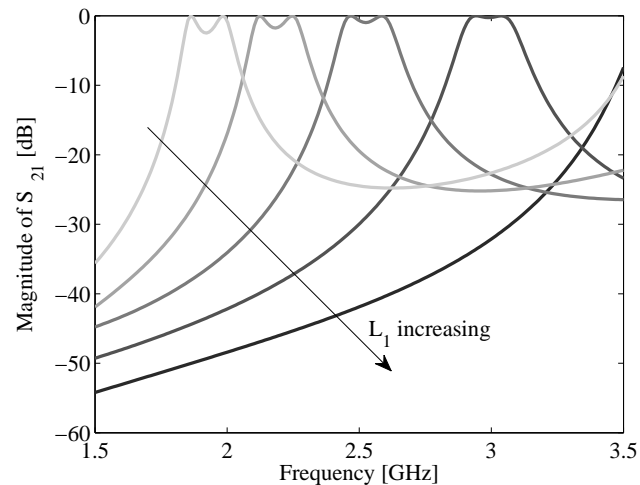


Figure 12. Hairpin Filter: Magnitude of S_{21} as a function of L_1 with scalable macromodel generated using *Algorithm II*.

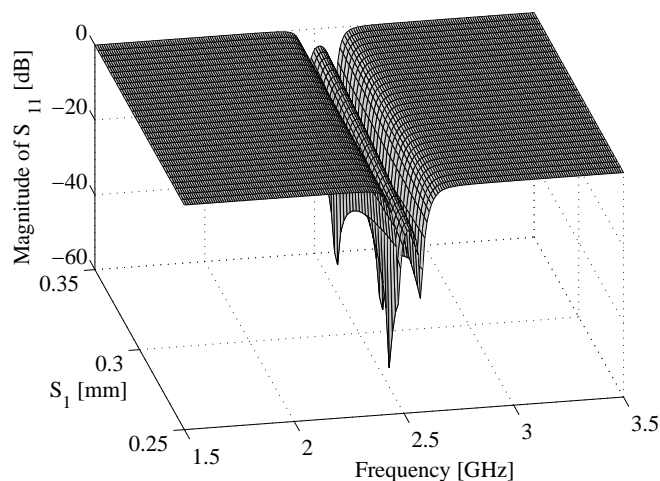


Figure 13. Hairpin Filter: Magnitude of S_{11} as a function of S_1 with scalable macromodel generated using *Algorithm II*.

To model the hairpin filter, *Algorithm I* and *Algorithm II* are used along with the scalable macromodeling method presented in [15]. The MAE error measure (2) is used to assess the accuracy of the models generated with a target accuracy of -50 dB. This it resulted in 513 design space points for *Algorithm II*, with an achieved accuracy of -50.67 dB, whereas *Algorithm I* did not converge even with 13071 points. As evident from these results, *Algorithm I* samples all the variables equally while *Algorithm II* selects the highly influential variables carefully, requiring much lesser points. This difference between the two algorithms becomes severe with higher dimensions.

Fig. 15 shows the design space points selected by *Algorithm II* respectively using a *parallel coordinates* plot [29]. In Fig. 15, the horizontal axis represents the four design variables S_1 , S_2 , L_1 and L_2 and the vertical axis represents their normalized values. Also, the black dots represent the sample points selected for each design variables and the gray dashed lines represent different design sample points. For instance, the bottom most horizontal line connecting the four black dots in Fig. 15 represent the design space point $[S_1, S_2, L_1, L_2] = [0, 0, 0, 0]$. In Fig. 15, the variable L_1 has the

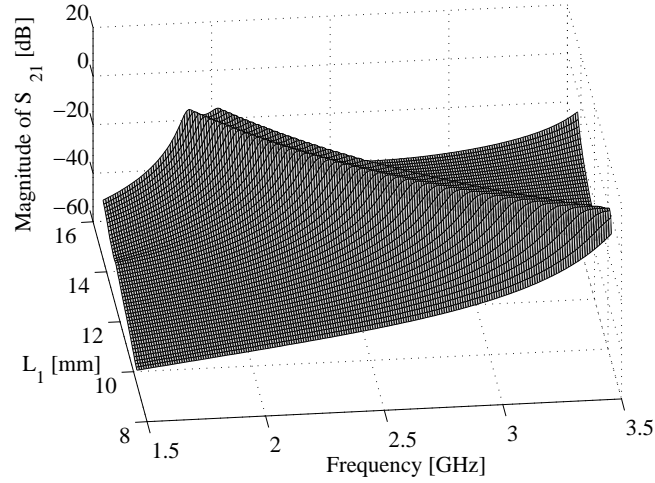


Figure 14. Hairpin Filter: Magnitude of S_{21} as a function of L_1 with scalable macromodel generated using *Algorithm II*.

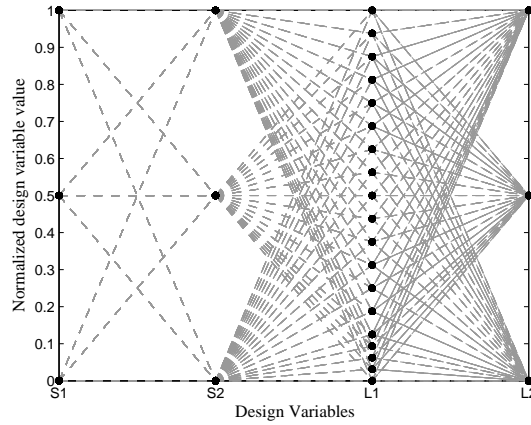


Figure 15. Hairpin Filter: Design space generated for Hairpin Filter using *Algorithm II*.

maximum number of samples selected on its axis using *Algorithm II*, as it is the most dynamic variable.

To illustrate the usefulness of the proposed method, the scalable macromodel generated with *Algorithm II* is used to optimize the bandpass filter. The specifications for the bandpass filter are given in terms of the scattering parameters S_{21} and S_{11} :

$$|S_{21}| > -2.5 \text{ dB} \quad \text{for} \quad 2.4 \text{ GHz} < \text{freq} < 2.5 \text{ GHz} \quad (4a)$$

$$|S_{11}| < -10 \text{ dB} \quad \text{for} \quad 2.4 \text{ GHz} < \text{freq} < 2.5 \text{ GHz} \quad (4b)$$

$$|S_{21}| < -40 \text{ dB} \quad \text{for} \quad \text{freq} < 1.7 \text{ GHz} \quad (4c)$$

$$|S_{21}| < -25 \text{ dB} \quad \text{for} \quad \text{freq} > 3.1 \text{ GHz}. \quad (4d)$$

The minimax optimization function *fminimax* in Matlab R2010a is used to perform an optimization with a cost function generated using the requirements of (4). Three optimization cases were considered with different starting conditions and are tabulated in Table II. The S-response calculated using the scalable macromodel is supplied to the minimax optimization routine, resulting in optimal design space points $[S_1^*, S_2^*, L_1^*, L_2^*]$ which satisfy the constraints in all three cases.

Table II. Hairpin Filter: Optimization results.

Initial Design Point (S_1, S_2, L_1, L_2) [mm]	Optimal Design Point ($S_1^*, S_2^*, L_1^*, L_2^*$) [mm]	# Function Evaluations	Optimal Cost	Processor Time [sec]
[0.30, 0.70, 12.00, 3.00]	[0.27, 0.75, 12.10, 3.25]	538	-8.4×10^{-4}	200.29
[0.34, 0.69, 14.10, 2.90]	[0.28, 0.75, 12.19, 3.20]	444	-6.3×10^{-4}	139.80
[0.33, 0.68, 11.50, 3.20]	[0.28, 0.75, 12.10, 3.24]	353	-10.9×10^{-4}	107.82

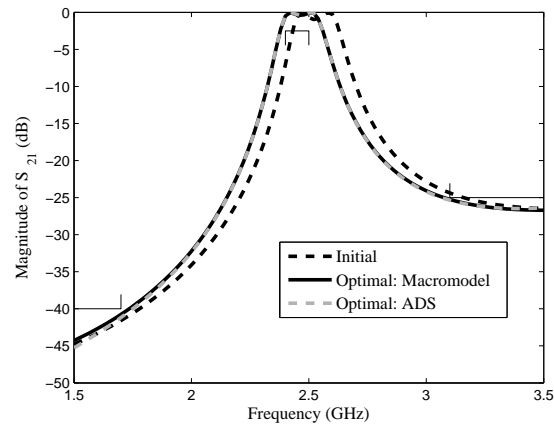
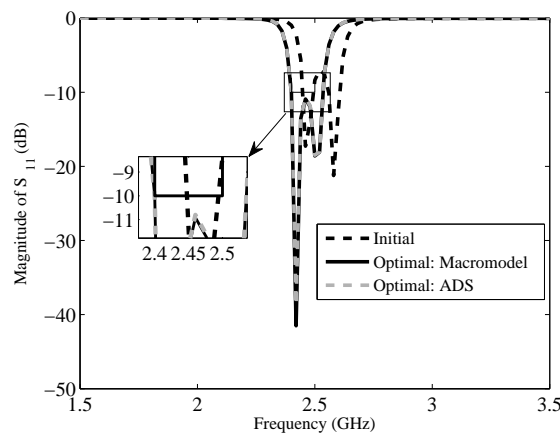
Figure 16. Hairpin Filter: Magnitude of S_{21} before and after optimization.Figure 17. Hairpin Filter: Magnitude of S_{11} before and after optimization.

Fig. 16 shows the magnitude of S_{21} for the second optimization case in the Table II. The actual data generated by the ADS Momentum software and that obtained using the scalable macromodel at the optimum design space point $[S_1^*, S_2^*, L_1^*, L_2^*]$ are shown in Fig. 16. As seen, both are in good agreement. The requirements (4) are shown by the thin black solid lines. Similar results are given for the magnitude of S_{11} in Fig. 17. As clearly seen, all the filter specifications are met for the optimal design point.

Since the cheap scalable macromodel is used in the minimax optimization, the CPU time per optimization is very small. This means that also global optimization schemes, which might require large number of function evaluations, can be used to avoid local optima without heavily increasing the CPU time. The generated scalable macromodel can also be used in other design activities such as sensitivity analysis, design space exploration, etc. Instead, if the ADS Momentum simulations had

been used for the optimization, the CPU time would have been very high. For instance, in case of the hairpin filter structure a single frequency sweep with 31 frequency samples requires approximately 145 seconds in ADS Momentum and 0.29 seconds with the generated scalable macromodel with around 500 times speed-up.

5. CONCLUSIONS

We have presented two sequential sampling algorithms for the macromodeling of parameterized system responses in model-based sequential sampling frameworks. Two sequential sampling algorithms for the automated generation of parametric macromodels have been discussed and investigated. All the proposed techniques have been validated and compared on pertinent numerical examples in terms of the number of points needed to cover the complete design space, modeling accuracy and CPU time. Also, one of the proposed algorithms has been used in the generation of a scalable macromodel for a microwave filter example with four design variables, and the generated scalable model is used in minimax optimization of the filter, validating the proposed sequential sampling method. However, it should be noted that these schemes work on grid-based design samples and hence susceptible to the curse of dimensionality when the number of design parameters increases.

REFERENCES

1. Peik S, Mansour R, Chow Y. Multidimensional Cauchy method and adaptive sampling for an accurate microwave circuit modeling. *IEEE Transactions on Microwave Theory and Techniques*, Dec 1998; **46**(12):2364–2371.
2. Lamecki A, Kozakowski P, Mrozowski M. Efficient implementation of the Cauchy method for automated CAD-model construction. *IEEE Microwave and Wireless Components Letters*, July 2003; **13**(7):268–270.
3. Cuyt A, Lenin R, Becuwe S, Verdonk B. Adaptive multivariate rational data fitting with applications in electromagnetics. *IEEE Transactions on Microwave Theory and Techniques*, May 2006; **54**(5):2265–2274.
4. Devabhaktuni V, Chattaraj B, Yagoub M, Zhang QJ. Advanced microwave modeling framework exploiting automatic model generation, knowledge neural networks, and space mapping. *IEEE Transactions on Microwave Theory and Techniques*, July 2003; **51**(7):1822–1833.
5. Lamecki A, Balewski L, Mrozowski M. Adaptive CAD-model construction schemes. *IEEE Transactions on Magnetics*, March 2009; **45**(3):1538–1541.
6. Basl P, Gohary R, Bakr M, Mansour R. Modelling of electromagnetic responses using a robust multi-dimensional Cauchy interpolation technique. *IET Microwaves, Antennas Propagation*, Nov 2010; **4**(11):1955–1964.
7. Lehmensiek R, Meyer P. Creating accurate multivariate rational interpolation models of microwave circuits by using efficient adaptive sampling to minimize the number of computational electromagnetic analyses. *IEEE Transactions on Microwave Theory and Techniques* Aug 2001; **49**(8):1419–1430.
8. Deschrijver D, Dhaene T, De Zutter D. Robust parametric macromodeling using multivariate orthonormal vector fitting. *IEEE Transactions on Microwave Theory and Techniques* July 2008; **56**(7):1661–1667.
9. Triverio P, Grivet-Talocia S, Nakhla M. A parameterized macromodeling strategy with uniform stability test. *IEEE Transactions on Advanced Packaging* Feb 2009; **32**(1):205–215.
10. Ferranti F, Knockaert L, Dhaene T. Parameterized S-parameter based macromodeling with guaranteed passivity. *IEEE Microwave and Wireless Component Letters* Oct 2009; **19**(10):608–610.
11. Ferranti F, Knockaert L, Dhaene T. Guaranteed passive parameterized admittance-based macromodeling. *IEEE Transactions on Advanced Packaging* Aug 2010; **33**(3):623–629.
12. Triverio P, Nakhla M, Grivet-Talocia S. Passive parametric macromodeling from sampled frequency data. *IEEE International Conference on Signal Propagation and Interconnects* May 2010; :117–120.
13. Ferranti F, Knockaert L, Dhaene T, Antonini G. Passivity-preserving parametric macromodeling for highly dynamic tabulated data based on Lur'e equations. *IEEE Transactions on Microwave Theory and Techniques* Dec 2010; **58**(12):3688–3696.
14. Deschrijver D, Crombecq K, Nguyen H, Dhaene T. Adaptive sampling algorithm for macromodeling of parameterized S-parameter responses. *IEEE Transactions on Microwave Theory and Techniques* Jan 2011; **59**(1):39–45.
15. Ferranti F, Knockaert L, Dhaene T, Antonini G. Parametric macromodeling based on amplitude and frequency scaled systems with guaranteed passivity. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* March/April 2012; **25**(2):139–151.
16. Montgomery D. *Design and analysis of experiments*. Student solutions manual, Wiley, 2008. URL <http://books.google.be/books?id=kMMJAm5bD34C>.
17. Crombecq K, Laermans E, Dhaene T. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. *European Journal of Operational Research* Nov 2011; **214**(3):683–696.
18. Crombecq K, Gorissen D, Deschrijver D, Dhaene T. A novel hybrid sequential design strategy for global surrogate modeling of computer experiments. *SIAM Journal on Scientific Computing* Aug 2011; **33**(4):1948–1974.

19. Rohrer R, Nosrati H. Passivity considerations in stability studies of numerical integration algorithms. *IEEE Transactions on Circuits and Systems*, Sep 1981; **28**(9):857 – 866.
20. Lewis D. Device model approximation using 2N trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Jan 1990; **9**(1):30 –38.
21. Fang GP, Yeh DC, Zweidinger D, Arledge LA, Gupta V. Fast, accurate MOS table model for circuit simulation using an unstructured grid and preserving monotonicity. *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, ASP-DAC '05, ACM: New York, NY, USA, 2005; 1102–1106.
22. Li XR, Zhao Z. Evaluation of estimation algorithms part I: incomprehensive measures of performance. *IEEE Transactions on Aerospace and Electronic Systems* Oct 2006; **42**(4):1340 –1358.
23. Weiser A, Zarantonello SE. A note on piecewise linear and multilinear table interpolation in many dimensions. *Mathematics of Computation* Jan 1988; **50**(181):189–196.
24. Gustavsen B, Semlyen A. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on Power Delivery* July 1999; **14**(3):1052–1061.
25. Gustavsen B. Improving the pole relocating properties of vector fitting. *IEEE Transactions on Power Delivery* July 2006; **21**(3):1587–1592.
26. Deschrijver D, Mrozowski M, Dhaene T, De Zutter D. Macromodeling of multiport systems using a fast implementation of the vector fitting method. *IEEE Microwave and Wireless Components Letters* June 2008; **18**(6):1587–1592.
27. Gustavsen B, Semlyen A. Fast passivity assessment for S-parameter rational models via a half-size test matrix. *IEEE Transactions on Microwave Theory and Techniques* Dec 2008; **56**(12):2701–2708.
28. Gustavsen B. Fast passivity enforcement for S-parameter models by perturbation of residue matrix eigenvalues. *IEEE Transactions on Advanced Packaging* Feb 2010; **33**(1):257–265.
29. Inselberg A. FT-1 the plane with parallel coordinates. *Parallel Coordinates*. Springer New York, 2009; 49–61.



Krishnan Chemmangat was born in Kerala, India, on December 28, 1984. Krishnan received the Bachelor of Technology (B.Tech) degree in Electrical and Electronics engineering from the University of Calicut, Kerala, India in 2006 and the Master of Technology (M.Tech) degree in Control and Automation from the Indian Institute of Technology Delhi, New Delhi, India in 2008. Since April 2010 he is active as a PhD student in the research group Internet Based Communication Networks and Services (IBCN) at Ghent University. His research interests include parametric macromodeling, sequential sampling and sensitivity analysis.



Francesco Ferranti received the B.S. degree in electronic engineering from the Universit degli Studi di Palermo, Palermo, Italy, in 2005, the M.S. degree in electronic engineering from the Universit degli Studi dell'Aquila, L'Aquila, Italy, in 2007, and the Ph.D. degree in electrical engineering from the University of Ghent, Ghent, Belgium, in 2011. He is currently a Post-Doctoral Research Fellow with in the research group Internet Based Communication Networks and Services (IBCN) at Ghent University. In September 2010, Francesco was awarded a research scholarship from the Scientific Research Fund Flanders (FWO) to cover a 4 month research visit to the Dept. of Electronics at Carleton University, Ottawa, Canada. There he worked on the model order reduction of delayed systems. He serves as a reviewer of several journals and conferences. His research interests include parametric macromodeling, parameterized model order reduction, modeling and simulation, non-linear systems, EMC/SI numerical modeling, system identification, efficient optimization



Tom Dhaene was born in Deinze, Belgium, on June 25, 1966. He received the Ph.D. degree in electrotechnical engineering from the University of Ghent, Ghent, Belgium, in 1993. From 1989 to 1993, he was Research Assistant at the University of Ghent, in the Department of Information Technology, where his research focused on different aspects of full-wave electro-magnetic circuit modeling, transient simulation, and time-domain characterization of high-frequency and high-speed interconnections. In 1993, he joined the EDA company Alphabit (now part of Agilent). He was one of the key developers of the planar EM simulator ADS Momentum. Since September 2000, he has been a Professor in the Department of Mathematics and Computer Science at the University of Antwerp, Antwerp, Belgium. Since October 2007, he is a Full Professor in the Department of Information Technology (INTEC) at Ghent University, Ghent, Belgium. As author or co-author, he has contributed to more than 250 peer-reviewed papers and

abstracts in international conference proceedings, journals and books about computational science and engineering, numerical analysis, and computer science. He is the holder of 5 U.S. patents.



Luc Knockaert received the M. Sc. Degree in physical engineering, the M. Sc. Degree in telecommunications engineering and the Ph. D. Degree in electrical engineering from Ghent University, Belgium, in 1974, 1977 and 1987, respectively. From 1979 to 1984 and from 1988 to 1995 he was working in North-South cooperation and development projects at the Universities of the Democratic Republic of the Congo and Burundi. He is presently affiliated with the Interdisciplinary Institute for BroadBand Technologies (www.ibbt.be) and a professor at the Dept. of Information Technology, Ghent University (www.intec.ugent.be). His current interests are the application of linear algebra and adaptive methods in signal estimation, model order reduction and computational electromagnetics. As author or co-author he has contributed to more than 120 international journal and conference publications. He is a member of MAA, SIAM and a senior member of IEEE.