

Reduced-Cost Microwave Filter Modeling Using a Two-Stage Gaussian Process Regression Approach

Jan Pieter Jacobs,¹ Slawomir Koziel²

¹Centre for Electromagnetism, Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, 0002, South Africa

²Engineering Optimization & Modeling Center, School of Science and Engineering, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland

ABSTRACT: A technique for the reduced-cost modeling of microwave filters is presented. Our approach exploits variable-fidelity electromagnetic (EM) simulations, and Gaussian process regression (GPR) carried out in two stages. In the first stage of the modeling process, a mapping between EM simulation filter models of low and high fidelity is established. The mapping is subsequently used in the second stage, making it possible for the final surrogate model to be constructed from training data obtained using only a fraction of the number of high-fidelity simulations normally required. As demonstrated using three examples of microstrip filters, the proposed technique allows us to reduce substantially (by up to 80%) the central processing unit (CPU) cost of the filter model setup, as compared to conventional (single-stage) GPR—the benchmark modeling method in this study. This is achieved without degrading the model generalization capability. The reliability of the two-stage modeling method is demonstrated through the successful application of the surrogates to surrogate-based filter design optimization.

Keywords: Gaussian processes; surrogate modeling; filter modeling; electromagnetic simulation; computer-aided design

1. Introduction

Full-wave electromagnetic (EM) analysis has become a primary design tool of contemporary microwave engineering. High-fidelity EM simulation ensures accurate design verification but it is normally computationally expensive. This might present a bottleneck to carrying out EM simulation-driven design tasks, such as parametric optimization, statistical analyses, and robust (e.g., tolerance-aware) design that require numerous evaluations of the structure of interest. This is particularly true not only for conventional optimization techniques, both gradient-based and derivative free [1, 2], but also for global optimization using metaheuristics (e.g., genetic algorithms [3, 4], and particle swarm optimizers [5]) that usually require a large number (thousands or more) of high-fidelity EM simula-

tions of the candidate geometries. Rapid development of computational tools and the presence of massive computing resources alleviate this difficulty to some extent only. This is because growing demand for accuracy (e.g., by including multiphysics and using finer discretization), as well as the necessity of accounting for various interactions and couplings between the structure of interest and its environment (e.g., antenna housing) tend to increase the computational complexity of the simulation process.

For the reasons elaborated above, fast and yet accurate models (also referred to as surrogates) become indispensable in microwave design. Perhaps the most popular way of creating such models is response surface approximation, where the surrogate is identified using a (training) set consisting of suitably selected input–output data samples. Properly trained response surface models exhibit good generalization capability, that is, allow for reliable prediction of the system/device response for the input parameter vectors that were not used in the training stage.

One of the most powerful approaches for constructing approximation-based surrogate models is Gaussian process

Correspondence to: J. P. Jacobs; e-mail: jppjacob@postino.up.ac.za.

regression (GPR) [6]. GPR has been shown to be a particularly effective modeling technique for performance characteristics such as antenna reflection coefficient and filter transmission coefficient responses against frequency [7–9]. Other kernel-based machine learning methods that have been used in this context include standard support vector regression, for example, [10], and the more flexible, GPR-based Bayesian support vector regression [11].

A Gaussian process (GP) is a stochastic process that generalizes the notion of Gaussian probability distribution to functions. Due to the Gaussian nature of the distribution, the calculations necessary in the learning and inference processes become tractable. Under suitable conditions, GPs can be considered equivalent to neural networks, but GPs are generally easier to implement and interpret—one reason is that training of far fewer parameters (in the order of the dimension of the input vectors) is required compared to the number of weights in a neural network (e.g., multilayer perceptron).

A drawback shared by response surface modelling techniques, including GPR (but also neural networks [12], support vector regression [13], and radial basis function interpolation [14]), is the usually large number of training points necessary to ensure sufficient predictive power of the model. In a typical setup, a few hundred to several thousand points might be required, which results in considerable computational overhead related to data acquisition, normally through massive EM simulations of the respective microwave structure. In many situations, such an overhead may be impractical or even prohibitive.

There have been various attempts to reduce the computational cost of creating the surrogate model setup cost. Adaptive sampling is perhaps the most popular approach, where the number of training data samples required to secure sufficient model accuracy is obtained in an iterative process consisting of model identification and subsequent generation of new samples using a suitable strategy. The strategies may be based on the actual modelling error at the testing locations [15], or the predicted error [16]. The methods described in [15] and [16] are developed to work as local ones, usually in an optimization context. Our focus is on the development of global surrogates (to be used as library models), where accurate prediction over the entire design space is of concern. Such models can be used in a wide range of applications, including parametric optimization or statistical analysis. Another approach to reduce the computational cost of surrogate model setup was presented in [17], where only the support vectors of an initial (global) Bayesian support vector regression model trained on low-fidelity data were selected for finely meshed simulation. A 31-to-48 percent reduction in model setup costs could be achieved without adversely affecting predictive ability.

In this work, we demonstrate that the central processing unit (CPU) cost of surrogate model setup can be substantially reduced by means of variable-fidelity EM simulations in combination with GPR. In particular, we propose a two-stage modeling scheme, where—in the first stage—we generate by full-wave simulation a low-fidelity (coarse)

training dataset of n points, and $n_{\text{aux}} < n$ points of the corresponding (computationally expensive) high-fidelity (fine) training set. These n_{aux} high-fidelity points are used to learn a model that maps low-fidelity training targets (either of $\text{Re}\{S_{21}\}$, $\text{Im}\{S_{21}\}$, or $|S_{21}|$) to the corresponding high-fidelity ones. The mapping is subsequently used to predict the remaining $n - n_{\text{aux}}$ high-fidelity targets that were not simulated. The n_{aux} simulated high-fidelity targets and the $n - n_{\text{aux}}$ predicted ones (together with the input vectors) form the n -point “approximate” high-fidelity training set. In the second stage, a final GPR model is constructed using the latter training set. The proposed two-stage modelling methodology exhibits substantial novelty with respect to the conventional GPR approach. It not only blends EM simulations of various level of fidelity into one surrogate model, but also does it in a parameter-less fashion (in contrast with, e.g., space mapping, SM [18–21]). While it has recently been applied with good effect to the modelling of antenna input characteristics [22], this study shows that the method is general enough to hold for filter responses too, which can be more challenging to model due to the sharp transitions explicitly required between, for example, passband and stopband portions of responses.

The article is organized as follows. In Section II we provide a brief theoretical overview of GPR following [6], and describe the two-stage modeling procedure. In Section III our approach is illustrated using three examples of microstrip filters. We demonstrate that exploitation of the knowledge embedded in the low-fidelity model allows for considerable (up to 80%) computational savings as compared to conventional GPR modelling, and without compromising modelling accuracy. Section IV demonstrates that the two-stage GPR surrogates are sufficiently accurate to be used for filter optimization within a surrogate-based optimization scheme involving output SM [19, 21]. Conclusions are presented in Section 5.

2. GPR modeling

A. Fundamentals of GPR

This section describes the statistics of GPR along the lines of [6], and explains how these equations map to filter modeling.

At the heart of GPR lies the multivariate Gaussian probability distribution. Consider n continuous random variables f_1, \dots, f_n with joint probability $p(f_1, \dots, f_n)$, or equivalently $p(\mathbf{f})$ with $\mathbf{f} = [f_1 \dots f_n]$. We assume that these variables are distributed according to the multivariate Gaussian (normal) distribution [6]:

$$p(\mathbf{f}|\mathbf{m}, A) = (2\pi)^{-n/2} |A|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{f}-\mathbf{m})^T A^{-1}(\mathbf{f}-\mathbf{m})\right) = N(\mathbf{m}, A) \quad (1)$$

where \mathbf{f} is a multidimensional “point” under the distribution; \mathbf{m} is the mean vector of length n ; and A is the covariance matrix of size $n \times n$. The shape of the distribution is determined by A (see, e.g., [23]).

Consider now regression with GPs (i.e., standard GPR). We are interested in learning a mapping between filter geometry dimensions and frequency, and $|S_{21}|$ (or possibly $\text{Re}/\text{Im}\{S_{21}\}$; for conciseness, we will refer to $|S_{21}|$ throughout). We start with a training dataset of n input–output pairs, $\{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$. The input vectors \mathbf{x}_i are of dimension P , whereas y_i denote the scalar target responses. Each input vector \mathbf{x}_i corresponds to a set of filter geometry parameters and a frequency value within the range of interest, and each output scalar y_i to the corresponding $|S_{21}|$ value. We also define a test dataset of n^* input–output pairs $\{(\mathbf{x}_i^*, y_i^*) \mid i = 1, \dots, n^*\}$, that is, input vectors \mathbf{x}_i^* consisting of previously unseen geometry-values-and-frequency for which we want to predict $|S_{21}|$, and output scalars y_i^* that are the associated values of $|S_{21}|$. These $|S_{21}|$ values are usually not available, but we generate them to test the model’s predictions.

Under GPR, we model the n training output scalars (associated with the n input vectors \mathbf{x}_i) by random variables $[f_1 \dots f_n]^T = [f(\mathbf{x}_1) \dots f(\mathbf{x}_n)]^T$, and the n^* test output scalars by random variables $[f_1^* \dots f_{n^*}^*]^T = [f(\mathbf{x}_1^*) \dots f(\mathbf{x}_{n^*}^*)]^T$, where $f(\mathbf{x})$ is a GP. A GP is a stochastic process that results from the generalization of the Gaussian probability distribution (1) to functions. This happens when the mean vector \mathbf{m} becomes infinitely long, that is, resulting in a mean function $m(\mathbf{x})$; and the two-dimensional covariance matrix A becomes infinitely large, with entries given by a covariance function $k(\mathbf{x}, \mathbf{x}')$. $f(\mathbf{x})$, which corresponds to an infinitely long vector, can be seen as a “point” under this distribution. The mean function has a standard statistical definition, namely

$$m(\mathbf{x}) = E[f(\mathbf{x})] \quad (2)$$

and the covariance function, which gives the covariance between outputs $f(\mathbf{x})$ and $f(\mathbf{x}')$ in terms of the associated inputs \mathbf{x} and \mathbf{x}' , is defined as [6]

$$k(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \quad (3)$$

where $E(X)$ is the expected value of the random variable X [note that the actual computation of covariance functions takes place through (7) and (8), as will be explained below]. Hence the GP $f(\mathbf{x})$ is a set consisting of an infinite number of random variables, of which any finite subset (e.g., the training outputs $\mathbf{f} = [f_1 \dots f_n]$) has a jointly Gaussian distribution by virtue of the general properties of the multivariate Gaussian distribution [6].

Predictions in the GP framework are carried out using straightforward probability rules applied to Gaussian multivariate distributions, for example, [6, Appendix A]. A jointly Gaussian distribution (1) with zero mean vector is assumed over the n training outputs and the n^* test outputs, therefore $n + n^*$ random variables in total. This is termed the prior distribution, and can be expressed as

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} \sim N \left(\mathbf{0}, \begin{bmatrix} K(X, X) & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix} \right) \quad (4)$$

Equation (4) states that the random variables contained in the vector $[\mathbf{f} \ \mathbf{f}^*]^T$ have a multivariate jointly Gaussian distribution with zero mean and covariance matrix $[\bullet]$. In (4), X is a matrix containing the training input vectors in its columns and X^* is a matrix containing the test input vectors; and $K(X, X^*)$ is a $n \times n^*$ submatrix of covariances evaluated between all possible pairs of n training and n^* test outputs—for example, $K_{12} = k(f(\mathbf{x}_1), f(\mathbf{x}_2^*)) = k(\mathbf{x}_1, \mathbf{x}_2^*)$. Other submatrices in the covariance matrix in (4) are similarly defined.

However, the training outputs \mathbf{y} are known, that is, $\mathbf{y} = [y_1 \dots y_n]$. Hence the distribution of the test outputs conditioned on the known training outputs \mathbf{y} can be computed—this is called the posterior distribution (again multivariate Gaussian), with mean vector \mathbf{p} and covariance matrix Σ [6]

$$\mathbf{p} = K(X^*, X)K(X, X)^{-1}\mathbf{y} \quad (5)$$

$$\Sigma = K(X^*, X^*) - K(X^*, X)K(X, X)^{-1}K(X, X^*) \quad (6)$$

The mean \mathbf{p} of the posterior distribution contains the $|S_{21}|$ predictions, that is, the most likely values of the test outputs associated with the test input vectors in X^* . In other words, $\mathbf{p} = [|S_{21}|_1 \ |S_{21}|_2 \ \dots \ |S_{21}|_{n^*}]$ where $|S_{21}|_1$ is the prediction for test input vector \mathbf{x}_1^* , $|S_{21}|_2$ is the prediction for \mathbf{x}_2^* , and so on. The diagonal of the covariance matrix Σ determines the corresponding predictive variances, which can be interpreted as the confidence of the model in its predictions.

The only type of parameterization in the model is through hyperparameters that determine the mean and covariance functions. Since the covariance functions determine the covariance matrices in the prior and posterior probability distributions [cf. (4) and (6)], they are crucial in determining the shapes of these distributions, and hence the GPs that will be favored by them. This study considers two covariance functions for calculating the covariance between two outputs $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$. The first is the squared-exponential covariance function with automatic relevance determination (ARD) [6],

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left[-\frac{1}{2} \sum_{k=1}^P \frac{(x_{i,k} - x_{j,k})^2}{\tau_k^2} \right] \quad (7)$$

In (1), $x_{i,k}$ and $x_{j,k}$ are the k th components of input vectors \mathbf{x}_i and \mathbf{x}_j , respectively ($k = 1, \dots, P$); $\tau_k > 0$ is the length-scale parameter that corresponds to component k of the two input vectors; and σ_f^2 is the signal variance. The length-scale parameters determine how quickly change occurs along particular dimensions of the input space. The second covariance function is the rational quadratic

function with ARD, which can be considered a scaled mixture of squared exponential functions with different length scales [6]:

$$k_{\text{RQ}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \left(1 + \frac{1}{2^\alpha} \sum_{k=1}^P \frac{(x_{i,k} - x_{j,k})^2}{\tau_k^2} \right)^{-\alpha} \quad (8)$$

In (8), $\alpha > 0$ is the shape parameter, with the remaining symbols defined as for (1). Together, σ_f^2 and the set of τ_k in (7) and (8)—as well as α in the case of (8)—constitute the hyperparameters of the covariance function.

Training in GPR entails optimizing the hyperparameters, which requires finding the set of hyperparameters that minimizes the negative log marginal likelihood, usually by means of gradient-based search. The log marginal likelihood is defined as [6]

$$\log p(\mathbf{y}|X) = -\frac{1}{2} \mathbf{y}^T K^{-1} \mathbf{y} - \frac{1}{2} \log |K| - \frac{n}{2} \log 2\pi \quad (9)$$

Here, K refers to the $n \times n$ matrix $K(X, X)$, with other symbols as defined previously. After optimization, the magnitude of hyperparameter τ_k will reflect the relative importance of the k th component of the input vectors: for example, large values of τ_k will ensure an insignificant contribution of that component to the covariance. This property is referred to as ARD.

The computational requirement for GPR is $O(n^3)$ due to the required inversion of $K(X, X)$ which is of dimension $n \times n$.

B. Two-Stage GPR Modeling

Our objective is to construct highly accurate GPR surrogate models \mathbf{R}_s that map geometry (design) variables and frequency to either (a) the separate real and imaginary parts, that is, $\text{Re}\{S_{21}\}$ and $\text{Im}\{S_{21}\}$ of S_{21} of the filter of interest, or (b) $|S_{21}|$ directly. Exploratory experiments revealed that Filters 1 and 2 (see Section 3) were best modeled using (a), while Filter 3 was better handled by (b). For the sake of conciseness, we will only refer to $|S_{21}|$ in the method description below.

Let us assume that to ensure sufficient model accuracy, an n -element set of high-fidelity training data (i.e., simulated using a finely discretized mesh) is necessary:

$$D_{\text{fine}} = \{(\mathbf{x}_i, y_{\text{fine},i}) | i=1, \dots, n\} \quad (10)$$

with P -dimensional input vectors

$$\mathbf{x}_i = [\mathbf{u}_i^T \ f_{oi}]^T = [u_{1i} \ u_{2i} \ \dots \ u_{Mi} \ f_{oi}]^T \quad (11)$$

and scalar targets $y_{\text{fine},i} = |S_{21}|_{\text{fine},i}$. The design vector $\mathbf{u}_i = [u_{1i} u_{2i} \dots u_{Mi}]^T$ consists of M adjustable geometry variables of the filter of interest and f_{oi} is the i th frequency sample

in the frequency band of interest; consequently, $P = M + 1$.

The cost of generating D_{fine} , however, may be prohibitively expensive. To address this problem, we adopt a two-stage modeling approach. It aims at setting up a final GPR model that is based on a fraction of the high-fidelity simulations required to set up D_{fine} but is almost as accurate as a GPR model trained on the actual D_{fine} . The details of the two modeling stages are as follows.

1. Two-Stage GPR: Stage One. The purpose of this stage is to “approximate” the expensive fine training dataset D_{fine} by a relatively inexpensive dataset $D_{\text{fine,approx}}$ of the same size; this is accomplished by means of a separate auxiliary GPR model \mathbf{R}_{aux} .

As a first step, we simulate—cheaply—the n input vectors of D_{fine} using a coarse discretization, yielding the dataset

$$D_{\text{coarse}} = \{(\mathbf{x}_i, y_{\text{coarse},i}) | i=1, \dots, n\} \quad (12)$$

with \mathbf{x}_i as before and $y_{\text{coarse},i} = |S_{21}|_{\text{coarse},i}$. In addition, we simulate (at high fidelity) a (small) randomly selected subset of D_{fine} consisting of $n_{\text{aux}} < n$ points. Using this subset of D_{fine} , we construct a training set D_{aux} for \mathbf{R}_{aux} as follows:

$$D_{\text{aux}} = \{(\mathbf{x}_{\text{aux},k}, y_{\text{fine},k}) | k=1, \dots, n_{\text{aux}}\} \quad (13)$$

where the $(M + 2)$ -dimensional training input vector

$$\mathbf{x}_{\text{aux},k} = [u_{1k} \ u_{2k} \ \dots \ u_{Mk} \ f_{ok} | S_{21}|_{\text{coarse},k}]^T \quad (14)$$

is of the form (7) augmented by the associated coarse $|S_{21}|$ target value from D_{coarse} , and the target $y_{\text{fine},k} = |S_{21}|_{\text{fine},k}$ is the corresponding $|S_{21}|$ value from the above subset of D_{fine} (recall that D_{coarse} and D_{fine} share the same set of input vectors; the only difference lies in the meshing density with which the targets have been obtained). Hence, we essentially learn a mapping between coarse and fine $|S_{21}|$ simulations using training data that correspond to n_{aux} specific instances of sets of design variables and frequency (the first $M + 1$ elements of the input vector $\mathbf{u}_{\text{aux},k}$ serves to uniquely identify the $|S_{21}|$ values). The aforementioned mapping represents the correlations between the coarse and fine model responses. Due to the fact that both models are physically related (as evaluated using the same EM solver), the mapping learned for a limited number of fine training points is likely to be preserved across the entire design space.

After training, we use \mathbf{R}_{aux} to predict, from their coarsely simulated counterparts, the $n - n_{\text{aux}}$ fine $|S_{21}|$ values that were not simulated; we refer to these predicted targets as $y_{\text{pred},k} = |S_{21}|_{\text{pred},k}$, $k = (n_{\text{aux}} + 1), \dots, n$. Taken together, the n_{aux} full-wave simulated fine $|S_{21}|$ target

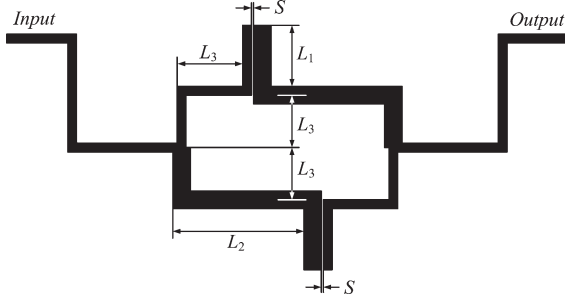


Figure 1 CCDBR filter: geometry [24].

values and the $n - n_{\text{aux}}$ predicted ones yield—along with input vectors consisting of geometry parameters and frequency of the form (7)—an n -point “approximate” fine training dataset for \mathbf{R}_s ,

$$D_{\text{fine,approx}} = \left\{ \begin{array}{l} (\mathbf{x}_k, y_{\text{fine},k}) \mid k=1, \dots, n_{\text{aux}} \\ (\mathbf{x}_k, y_{\text{pred},k}) \mid k=(n_{\text{aux}}+1), \dots, n \end{array} \right\} \quad (15)$$

Obtaining the targets $y_{\text{pred},k}$ via model predictions (as opposed to direct full-wave simulations) can result in significant savings in computational costs, as will be outlined below.

2. Two-Stage GPR: Stage Two. In the second stage, we use $D_{\text{fine,approx}}$ (instead of the full D_{fine} , which is not available) as training set for \mathbf{R}_s , the desired final surrogate that maps design variables and frequency to $|S_{21}|$, using the “conventional” GPR of Section II.A In Section IV, we show that these surrogates are sufficiently accurate to be used to good effect for optimization using SM.

It is worth noting that the reduction in the number of simulated high-fidelity training points without negatively affecting model accuracy is possible because we exploit the knowledge embedded in the low-fidelity model. In our approach, the use of this knowledge is implemented through the mapping learned in the first stage that identifies correlations between the low- and high-fidelity simulation data.

3. Verification examples

In this section, we present numerical verification of the two-stage GPR modeling technique using three examples of microstrip filters.

A. Capacitively Coupled Dual-Behavior Resonator Filter (Filter 1)

Consider the second-order capacitively coupled dual-behavior resonator (CCDBR) microstrip filter [24], see Figure 1. The vector of design variables is $\mathbf{u} = [L_1 \ L_2 \ L_3]^T$. Microstrip line widths were 0.25 and 0.5 mm, S was 0.05 mm, the substrate parameters were: $h = 0.254$ mm, and $\epsilon_r = 9.9$. The region of interest for

setting up the surrogate model was defined as $[\mathbf{u}^0 - \delta, \mathbf{u}^0 + \delta]$ with $\mathbf{u}^0 = [3 \ 5 \ 1.5]^T$ mm, and $\delta = [1 \ 1 \ 0.5]^T$ mm. The goal is to model $|S_{21}|$ for the frequency range of 2–6 GHz. The rational quadratic covariance function (8) was used during both stages of our two-stage approach.

For this example, the training data input vectors for creating our surrogate model \mathbf{R}_s were allocated using Latin hypercube sampling (LHS) [25]. For each training geometry, 12 frequencies were (uniformly) randomly sampled within the frequency range 2–6 GHz. As a result, a different set of frequencies was effectively assigned with each geometry. The total number of training vectors for \mathbf{R}_s was $n = 600 \times 12 = 7200$; training input vectors had the form $\{\mathbf{x}_i = [\mathbf{u}_i^T \ f_{oi}]^T \mid i = 1, \dots, n\}$. As before, f_{oi} were frequency values. The testing set consisted of 50 separate geometries, also obtained through LHS, however, using 81 frequencies per geometry, distributed uniformly over the frequency range of interest ($n^* = 4050$).

The (full) n -point high-fidelity training dataset D_{fine} was obtained from simulations in FEKO [26], that is, the high-fidelity model \mathbf{R}_f . Hence D_{fine} consisted of n input–output pairs, $\{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$, with $\mathbf{x}_i = [\mathbf{u}_i^T \ f_{oi}]^T = [L_1 \ L_2 \ L_3 \ f_{oi}]^T$, and $y_i = \text{Re}\{S_{21}\}_{\text{fine},i}$ or $\text{Im}\{S_{21}\}_{\text{fine},i}$. Similarly, the training set D_{coarse} was obtained from FEKO simulations at a low mesh density (i.e., the low-fidelity model \mathbf{R}_c). Total mesh numbers for \mathbf{R}_f and \mathbf{R}_c were 614 (evaluation time 6 s per frequency) and 130 (evaluation time 0.3 s per frequency), respectively. The test inputs were only simulated at the fine mesh density, yielding the test dataset D_{test} used to evaluate the predictive capabilities of \mathbf{R}_s .

The first stage of the proposed modelling approach (carried out separately for each of $\text{Re}\{S_{21}\}$ and $\text{Im}\{S_{21}\}$ —for conciseness, we refer to only $\text{Re}\{S_{21}\}$) involves construction of the training set D_{aux} . This was realized by randomly selecting n_{aux} data points from the original set D_{fine} . The auxiliary surrogate \mathbf{R}_{aux} was then set up as described in Section 2.2. In particular, the training set D_{aux} consisted of n_{aux} input–output pairs $\{(\mathbf{x}_{\text{aux},k}, y_{\text{fine},k}) \mid k = 1, \dots, n_{\text{aux}}\}$, with $\mathbf{x}_{\text{aux},k} = [\mathbf{u}_k^T \ f_{ok} \ \text{Re}\{S_{21}\}_{\text{coarse},k}]^T = [L_1 \ L_2 \ L_3 \ f_{ok} \ \text{Re}\{S_{21}\}_{\text{coarse},k}]^T$, and $y_{\text{fine},k} = \text{Re}\{S_{21}\}_{\text{fine},k}$. After training the model by minimizing the negative of the log marginal likelihood [cf. (9)], the model was used to estimate the rest of the high-fidelity target values in D_{fine} by finding the mean of the posterior distribution [cf. (5)]—that is, yielding $n - n_{\text{aux}}$ $\text{Re}\{S_{21}\}_{\text{pred}}$ values. This was done for $n_{\text{aux}}/n \times 100\% \in \{40\%, 30\%, 20\%\}$, and the predictive errors of \mathbf{R}_{aux} (i.e., the root mean square values of the $n - n_{\text{aux}}$ residuals of $\text{Re}\{S_{21}\}_{\text{pred}}$ and $\text{Re}\{S_{21}\}_{\text{fine}}$) are listed in Table I for each case.

Next, we constructed “approximate” fine training datasets $D_{\text{fine,approx}}$ [as described by (15)] corresponding to $n_{\text{aux}}/n \times 100\% \in \{40\%, 30\%, 20\%\}$, and trained GPR models \mathbf{R}_s on each set (the second stage of our method). The n -point “approximate” fine training set corresponding to a specific n_{aux} value was

TABLE I Predictive Errors^a of Auxiliary Filter Models \mathbf{R}_{aux} on Remaining $n - n_{\text{aux}}$ Fine Training Data Points

$n_{\text{aux}}/n \times$ 100%	RMSE (%)				
	Filter 1		Filter 2		Filter 3
	$\text{Re}\{S_{21}\}$	$\text{Im}\{S_{21}\}$	$\text{Re}\{S_{21}\}$	$\text{Im}\{S_{21}\}$	$ S_{21} $
40	1.45	1.47	1.79	2.03	3.17
30	1.77	1.63	1.73	1.98	3.43
20	1.87	1.97	2.74	2.23	3.75

^aNormalized root mean square error (RMSE), expressed as a percentage of the target value range.

$$D_{\text{fine,approx}} = \left\{ \begin{array}{l} ([L_{1k} \ L_{2k} \ L_{3k} \ f_{ok}], \text{Re}\{S_{21}\}_{\text{fine},k}) \mid k=1, \dots, n_{\text{aux}} \\ ([L_{1k} \ L_{2k} \ L_{3k} \ f_{ok}], \text{Re}\{S_{21}\}_{\text{pred},k}) \mid k=(n_{\text{aux}}+1), \dots, n \end{array} \right\}$$

with $n = 7200$ as noted above.

The predictive errors of these models on the test dataset D_{test} are listed in Table II. The test set, which was independent of the training set, was given by

$$D_{\text{test}} = \{[L_{1j} \ L_{2j} \ L_{3j} \ f_{oj}]^T, \text{Re}\{S_{21}\}_{\text{fine},j} \mid j=1, \dots, n^*\}$$

with $n^* = 4050$.

For the sake of comparison, the predictive error for the case where the full D_{fine} was used as training data, that is, $n_{\text{aux}}/n \times 100\% = 100\%$ is also indicated. We refer to this model as $\mathbf{R}_{\text{s,full}}$. Predictive accuracies appeared to be good given the relatively small proportions of high-fidelity data present in the ‘‘approximate’’ fine training datasets.

B. Second-Order Ring-Resonator Bandpass Filter (Filter 2)

Consider the second-order ring resonator bandpass filter [27] shown in Figure 2. The design parameters were $\mathbf{u} = [L_1 \ L_2 \ S_1 \ W_1 \ W_2]^T$, while the substrate height and dielectric constant were 1.52 mm and 4.32, respectively. The design space was $[\mathbf{u}^0 - \boldsymbol{\delta}, \mathbf{u}^0 + \boldsymbol{\delta}]$ with $\mathbf{u}^0 = [20 \ 22 \ 0.2 \ 0.8 \ 1.7]^T$ mm, and $\boldsymbol{\delta} = [2 \ 2 \ 0.1 \ 0.1 \ 0.1]^T$ mm. The objective was to model $|S_{21}|$ for the frequency range of 1–3 GHz.

TABLE II Predictive Errors of Surrogate Filter Models \mathbf{R}_{s} on Fine Test Data

$n_{\text{aux}}/n \times$ 100%	RMSE (%)				
	Filter 1		Filter 2		Filter 3
	$\text{Re}\{S_{21}\}$	$\text{Im}\{S_{21}\}$	$\text{Re}\{S_{21}\}$	$\text{Im}\{S_{21}\}$	$ S_{21} $
100 ($\mathbf{R}_{\text{s,full}}$) ^a	3.11	3.22	2.56	2.93	3.92
40	3.14	3.31	3.12	3.37	4.12
30	3.18	3.38	2.99	3.33	4.22
20	3.24	3.36	3.54	3.53	4.39

^a $\mathbf{R}_{\text{s,full}}$ is the benchmark that we compare to, that is, standard GPR using D_{fine} as training data (the fine training data obtained in full via EM simulations).

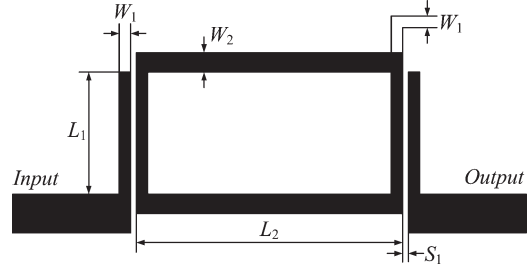


Figure 2 Second-order ring resonator bandpass filter: geometry [27].

The setup of models \mathbf{R}_{aux} and \mathbf{R}_{s} was similar to that used for Filter 1, with the exception that the squared-exponential covariance function (7) was used throughout. Tables I and II show the predictive errors for these models, confirming that both types of models had good predictive capabilities. As in the case of Filter 1, predictive accuracies of in particular the final surrogate \mathbf{R}_{s} appeared to be good given the relatively small fractions of high-fidelity data present in the ‘‘approximate’’ fine training datasets.

For this example, training data consists of 400 geometries allocated using LHS. Six randomly selected frequencies per geometry were used, which results in the total number of training points of $n = 2400$. For the testing purposes, we use 50 additional geometries (also LHS-allocated), with 81 frequencies per geometry (uniformly distributed). Both the high- and low-fidelity models were simulated in FEKO [26]. Total mesh numbers for \mathbf{R}_{f} and \mathbf{R}_{c} were 828 (evaluation time 8 s per frequency) and 64 (evaluation time about 0.1 s per frequency), respectively.

C. Fourth-Order Ring-Resonator Bandpass Filter (Filter 3)

The geometry of the filter [27] is shown in Figure 3. For this example, the design parameters were $\mathbf{u} = [L_1 \ L_2 \ L_3 \ S_1 \ S_2]^T$. The microstrip widths W_1 and W_2 were fixed to 1.2 and 0.8 mm, respectively. The substrate height and dielectric constant were 1.52 mm and 4.32. The surrogate model was set up over the interval $[\mathbf{u}^0 - \boldsymbol{\delta}, \mathbf{u}^0 + \boldsymbol{\delta}]$ with $\mathbf{x}^0 = [24$

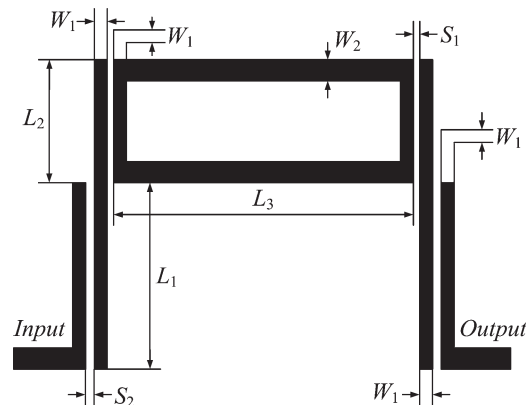


Figure 3 Fourth-order ring resonator bandpass filter: geometry [27].

20 26 0.1 0.1]^T mm, and $\delta = [4 4 4 0.05 0.05]^T$ mm. The objective was to model $|S_{21}|$ over the frequency band 1–3 GHz.

For this last example, training data consisted of 600 LHS-allocated geometries. We used 16 frequencies per geometry (randomly selected) so that the total number of training points was $n = 9600$. The test set contained 50 additional geometries; 81 equally spaced frequencies per geometry were used. Both filter models (high- and low-fidelity) were simulated in FEKO [26]. Total mesh numbers for \mathbf{R}_f and \mathbf{R}_c were 916 (evaluation time 1.0 s per frequency) and 174 (evaluation time 0.5 s per frequency), respectively.

The models \mathbf{R}_{aux} and \mathbf{R}_s were setup in the same way as before using the methodology of Section II.B [in both instances the rational quadratic covariance function (8) was used]. As was noted in Section 2.2, the magnitude of S_{21} was modeled directly for this filter. Tables I and II give the respective predictive errors, demonstrating as in the case of Filters 1 and 2 that both types of models could make good predictions.

D. Discussion

The results presented in the previous section indicate that combining low- and high-fidelity EM simulation data using the proposed modeling technique allows for very significant reduction—compared to conventional GPR modeling—of the computational overhead related to training data acquisition without compromising the predictive power of the surrogate model.

It should be emphasized that modeling of filter characteristics is a difficult task because of the high nonlinearity of the responses. In practice, this can be effectively realized only if the number of design variables is relatively low, or the design space is small (in terms of the range of variability of the geometry/material parameters). Thus, the presented examples are quite representative of what can be accomplished using approximation techniques such as GPR. For the sake of illustration, we focused on modeling the transmission characteristics of the filters. Modeling of return loss would be actually much more challenging mostly due to the nonlinearity issues.

4. Application examples: Filter optimization

As application examples, the three filter structures considered in Section 3 were optimized using the final GPR surrogate models \mathbf{R}_s . This can be considered a practical way of validating the proposed modeling technique. It should be emphasized that the GPR surrogates considered in this article are supposed to be multiple-purpose library models. Filter optimization with respect to various sets of design specifications is one example of a typical application task. Another could be robust (yield-driven) optimization or statistical analysis.

The design specifications and initial designs were the following:

- CCDBR filter: $|S_{21}| \geq -3$ dB for 3.8–4.2 GHz and $|S_{21}| \leq -20$ dB for 2–3.2 GHz and for 4.8–6 GHz; initial design: $\mathbf{u}_{\text{init}} = [3.5 4.5 1.5]^T$ mm;

TABLE III Filter Optimization Results

Filter	Model	Minimax Specification	
		Error at Final Design ^a (dB)	Optimization Cost ^b
1	$\mathbf{R}_{s,\text{full}}$	−2.4	3
	\mathbf{R}_s (40%) ^c	−2.3	4
	\mathbf{R}_s (30%) ^c	−2.2	3
	\mathbf{R}_s (20%) ^c	−2.4	3
2	$\mathbf{R}_{s,\text{full}}$	−0.0	4
	\mathbf{R}_s (40%) ^c	−0.1	4
	\mathbf{R}_s (30%) ^c	−0.0	4
	\mathbf{R}_s (20%) ^c	−0.1	4
3	$\mathbf{R}_{s,\text{full}}$	−0.2	4
	\mathbf{R}_s (40%) ^c	−0.3	3
	\mathbf{R}_s (30%) ^c	−0.4	2
	\mathbf{R}_s (20%) ^c	−0.3	3

^aMaximum violation of $|S_{21}|$ specifications at the frequency bands of interest.

^bNumber of evaluations of the high-fidelity model \mathbf{R}_f , including evaluation at the initial design.

^c40% refers to the model trained on $D_{\text{fine,approx}}$ using 40% data points that are actually simulated at high fidelity (30 and 20% accordingly).

- Second-order ring resonator filter: $|S_{21}| \geq -1$ dB for 1.8–2.2 GHz and $|S_{21}| \leq -20$ dB for 1–1.55 GHz and for 2.45–3 GHz; initial design $\mathbf{u}_{\text{init}} = [18.0 22.0 0.2 0.8 1.7]^T$ mm;
- Fourth-order ring resonator filter: $|S_{21}| \geq -1$ dB for 1.75–2.25 GHz and $|S_{21}| \leq -20$ dB for 1–1.5 GHz and for 2.5–3 GHz; initial design $\mathbf{u}_{\text{init}} = [25 20 25 0.1 0.1]^T$ mm.

The first stage of the optimization process is to optimize the surrogate model \mathbf{R}_s . Due to nonzero modeling error, an iterative design refinement procedure is used that employs SM technology [19]

$$\mathbf{u}^{(i+1)} = \arg \min_{\mathbf{x}} U(\mathbf{R}_{\text{su}}^{(i)}(\mathbf{u})) \quad (16)$$

where $\mathbf{R}_{\text{su}}^{(i)}$ denotes a surrogate model, here, constructed by means of an additive response correction (output SM) [19]. The function U encoded the design specifications. Here, the design specifications are formulated so that the problem is, in effect, a minimax optimization one (i.e., one attempts to minimize the maximum violation of the design specifications over the frequency band(s) of interest). For simplicity, the symbol \mathbf{R}_{co} is used to denote either of $\mathbf{R}_{s,\text{full}}$ or \mathbf{R}_s , which can be considered the underlying “coarse” models in the SM terms. The surrogate model is defined using the response of the high-fidelity model \mathbf{R}_f at $\mathbf{u}^{(i)}$ as follows

$$\mathbf{R}_{\text{su}}^{(i)}(\mathbf{u}) = \mathbf{R}_{\text{co}}(\mathbf{u}) + \mathbf{d}^{(i)} \quad (17)$$

with

$$\mathbf{u}^{(i)} = \mathbf{R}_f(\mathbf{u}^{(i)}) - \mathbf{R}_{co}(\mathbf{u}^{(i)}) \quad (18)$$

The additive correction term $\mathbf{d}^{(i)}$ is calculated to ensure zero-order consistency (i.e., $\mathbf{R}_{su}^{(i)}(\mathbf{u}^{(i)}) = \mathbf{R}_f(\mathbf{u}^{(i)})$) between the surrogate and the high-fidelity model \mathbf{R}_f [28] at the current design $\mathbf{x}^{(i)}$. The use of response correction is necessary because of the presence of some residual inaccuracy (misalignment between the surrogate model and the high-fidelity one). In practice, because of good initial accuracy of the GPR surrogates, one or two iterations of the algorithm (16) are usually sufficient to yield an optimized design. It should be noted that each iteration (16) requires only one evaluation of the high-fidelity model: the cost of optimizing the surrogate model can be neglected as compared to \mathbf{R}_f evaluation.

The optimization results are presented in Table III, as well as in Figures 4 and 5. Figure 4 shows, for Filters 1 through 3, the responses of models $\mathbf{R}_{s,\text{full}}$ and \mathbf{R}_f (i.e., direct high-fidelity FEKO simulations) at the initial designs, and the high-fidelity model response at the final designs. Similarly, Figure 5 shows the GPR surrogate

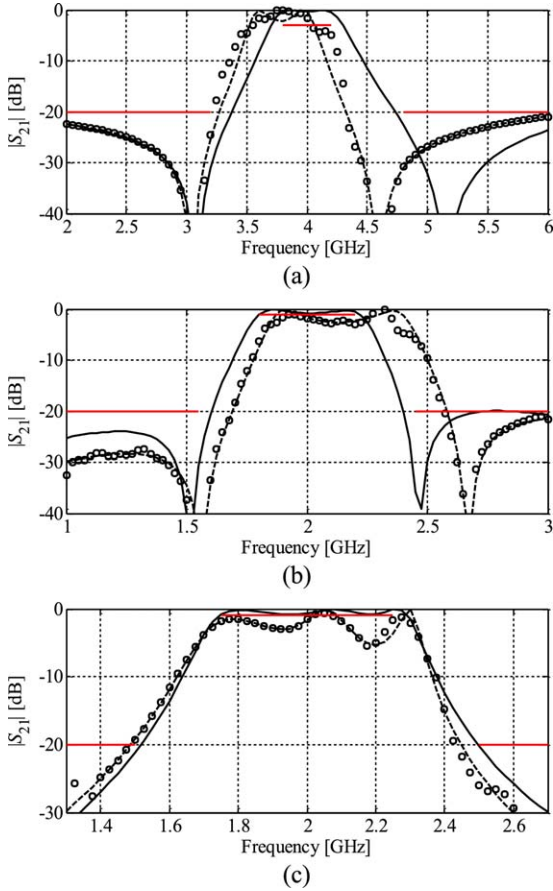


Figure 4 Optimization results for the three filter structures: responses of $\mathbf{R}_{s,\text{full}}$ (o) and \mathbf{R}_f (- -) at the initial design, and \mathbf{R}_f at the optimized design (—) for (a) Filter 1, (b) Filter 2, and (c) Filter 3. Horizontal solid lines denote design specifications.

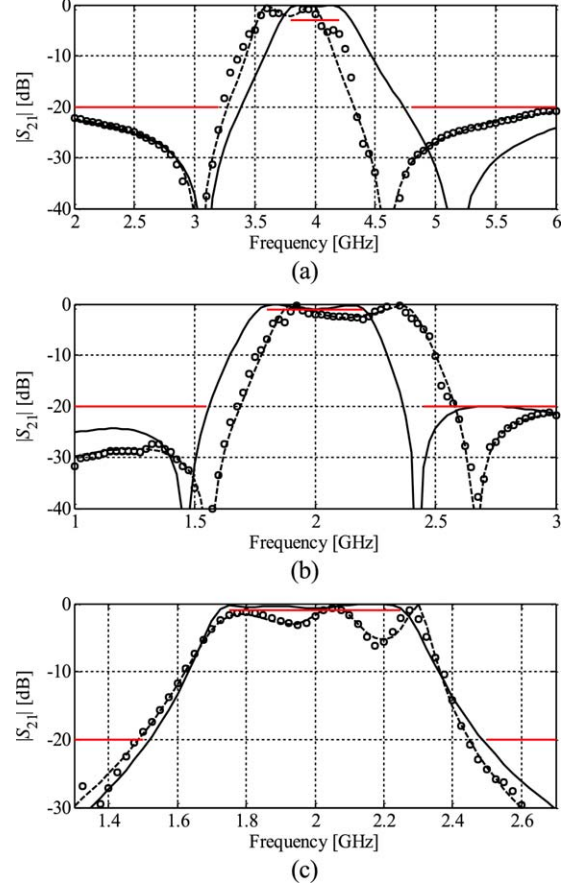


Figure 5 Optimization results for the three filter structures: responses of \mathbf{R}_s (o) and \mathbf{R}_f (- -) at the initial design, and \mathbf{R}_f at the optimized design (—) for (a) Filter 1, (b) Filter 2, and (c) Filter 3. Horizontal lines denote design specifications. The \mathbf{R}_s models represented here were trained on $D_{\text{fine,approx}}$ utilizing 20% data points simulated at high fidelity.

model responses trained on the “approximate” high-fidelity dataset $D_{\text{fine,approx}}$ (here with $n_{\text{aux}}/n \times 100\% = 20\%$) and the \mathbf{R}_f model responses at $\mathbf{x}^{(0)}$, and the response of \mathbf{R}_f at the final designs.

The numerical results are summarized in Table III. It is worth noting that the differences between the design quality and cost (the latter expressed in terms of number of evaluations of the high-fidelity model \mathbf{R}_f), are very small for the GPR models obtained using the original and approximate high-fidelity training datasets. The average design cost is about three evaluations of the high-fidelity model and it is similar in all cases, regardless of the n_{aux}/n ratio

5. Conclusion

A method has been presented for the efficient modeling of highly challenging nonlinear microwave filter responses using variable-fidelity EM simulations and GPR within a two-stage framework. The knowledge embedded in the low-fidelity filter simulations is used in

the first modeling stage, where the mapping between the EM models of different fidelity is established. This leads to substantial reduction of the number of high-fidelity simulations necessary to construct—in the second stage—the surrogate model of the filter of interest. These computational savings are achieved without compromising the predictive power of the final surrogate. Our approach has been verified using three examples of microstrip filters with satisfactory results obtained even for very limited high-fidelity simulations, corresponding to only 20–40% of what is required in setting up high-fidelity training sets for conventional GP modeling. The two-stage surrogates have been also successfully used for filter optimization, proving their usefulness in the design context.

Acknowledgment

This work was supported in part by the Icelandic Centre for Research (RANNIS) Grants 120016021 and 13045051.

References

1. J. Nocedal and S.J. Wright, *Numerical optimization*, Springer series in operations research, Springer, New York, 2000.
2. A.R. Conn, K. Scheinberg, and L.N. Vicente, *Introduction to derivative-free optimization*, MPS-SIAM series on Optimization, MPS-SIAM, Philadelphia, 2009.
3. R.L. Haupt, Antenna design with a mixed integer genetic algorithm, *IEEE Trans Antennas Propag* 55 (2007), 577–582.
4. M.F. Pantoja, P. Meincke, and A.R. Bretones, A hybrid genetic algorithm space-mapping tool for the optimization of antennas, *IEEE Trans Antennas Propag* 55 (2007), 777–781.
5. N. Jin and Y. Rahmat-Samii, Parallel particle swarm optimization and finite-difference time-domain (PSO/FDTD) algorithm for multiband and wide-band patch antenna designs, *IEEE Trans Antennas Propag* 53 (2005), 3459–3468.
6. C.E. Rasmussen and C.K.I. Williams, *Gaussian processes for machine learning*. MIT Press, Cambridge, MA, 2006.
7. J.P. De Villiers and J.P. Jacobs, Gaussian process modeling of CPW-fed slot antennas, *Progr Electromagn Res* 98 (2009), 233–249.
8. J.P. Jacobs and J.P. De Villiers, Gaussian-process-regression-based design of ultrawide-band and dual-band CPW-fed slot antennas, *J Electromagn Waves Appl* 24 (2010), 1763–1772.
9. J.P. Jacobs and S. Koziel, Gaussian process modeling of multi-variable microwave filters using non-standard covariance functions, *International Review of Progress in Applied Computational Electromagnetics*, Columbus, OH, April 10–14, 2012, pp. 1082–1086.
10. G. Angiulli, M. Cacciola, and M. Versaci, Microwave devices and antennas modelling by support vector regression machines, *IEEE Trans Magn* 43 (2007), 1589–1592.
11. J.P. Jacobs, Bayesian support vector regression with automatic relevance determination Kernel for modeling of antenna input characteristics, *IEEE Trans Antennas Propag* 60 (2012), 2114–2118.
12. H. Kabir, Y. Wang, M. Yu, and Q.J. Zhang, Neural network inverse modeling and applications to microwave filter design, *IEEE Trans Microwave Theory Tech* 56 (2008), 867–879.
13. L. Xia, J. Meng, R. Xu, B. Yan, and Y. Guo, Modeling of 3-D vertical interconnect using support vector machine regression, *IEEE Microwave Wireless Compon Lett* 16 (2006), 639–641.
14. M.D. Buhmann and M.J. Ablowitz, *Radial basis functions: Theory and implementations*, Cambridge University Press, Cambridge, UK, 2003.
15. V.K. Devabhaktuni, M.C.E. Yagoub, and Q.J. Zhang, A robust algorithm for automatic development of neural network models for microwave applications, *IEEE Trans Microwave Theory Tech* 49 (2001), 2282–2291.
16. I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, Surrogate-based infill optimization applied to electromagnetic problems, *Int J RF Microwave Comput-Aided Eng* 20 (2010), 492–501.
17. J.P. Jacobs, S. Koziel, and S. Ogurtsov, Computationally efficient multi-fidelity Bayesian support vector regression modeling of planar antenna input characteristics, *IEEE Trans Antennas Propag* 61 (2013), 980–984.
18. J.W. Bandler, Q.S. Cheng, S.A. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, and J. Søndergaard, Space mapping: The state of the art, *IEEE Trans Microwave Theory Tech* 52 (2004), 337–361.
19. S. Koziel, J.W. Bandler and K. Madsen, A space mapping framework for engineering optimization: Theory and implementation, *IEEE Trans Microwave Theory Tech* 54 (2006), 3721–3730.
20. M.B. Yelten, T. Zhu, S. Koziel, P.D. Franzon, and M.B. Steer, Demystifying surrogate modeling for circuits and systems, *IEEE Circuits Syst Mag* 12 (2012), 45–63.
21. Q.S. Cheng, J.W. Bandler, S. Koziel, M.H. Bakr, and S. Ogurtsov, The state of the art of microwave CAD: EM-based optimization and modeling, *Int J RF Microwave Comput-Aided Eng* 20 (2010), 475–491.
22. J.P. Jacobs and S. Koziel, Two-stage framework for efficient Gaussian process modeling of antenna input characteristics, *IEEE Trans Antennas Propag* 62 (2014), 706–713.
23. R.A. Johnson and D.W. Wichern, *Applied multivariate statistical analysis*, 6th ed., Pearson, Upper Saddle River, NJ, 2007.
24. A. Manchec, C. Quendo, J.-F. Favennec, E. Rius, and C. Person, Synthesis of capacitive-coupled dual-behavior resonator (CCDBR) filters, *IEEE Trans Microwave Theory Tech* 54 (2006), 2346–2355.
25. B. Beachkofski and R. Grandhi, Improved distributed hypercube sampling, *American Institute of Aeronautics and Astronautics*, paper AIAA 2002-1274, 2002.
26. FEKO, Suite 5.3, EM Software & Systems-S.A. (Pty) Ltd, 32 Techno Lane, South Africa, 2008.
27. M.H.M. Salleh, G. Prigent, O. Pigaglio, and R. Crampagne, Quarter-wavelength side-coupled ring resonator for bandpass filters, *IEEE Trans Microwave Theory Tech* 56 (2008), 156–162.
28. N.M. Alexandrov and R.M. Lewis, An overview of first-order model management for engineering optimization, *Optim Eng* 2 (2001), 413–430.

Biographies



Pieter Jacobs holds B.E., M.E., and Ph.D. degrees in electronic engineering from the University of Pretoria, South Africa, and a doctorate in music from Yale University. His research interests include modeling and optimization of microwave antennas and devices, computational electromagnetics, slot antennas and arrays, and pattern recognition in music. He is at present a Senior Lecturer in the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria.



Slawomir Koziel received the M.Sc. and Ph.D. degrees in electronic engineering from Gdansk University of Technology, Poland, in 1995 and 2000, respectively. He also received the M.Sc. degrees in theoretical physics and in mathematics, in 2000 and 2002, respectively, as well as the Ph.D. in mathematics in 2003, from the University of Gdansk, Poland. He is currently a Professor with the School of Science and Engineering, Reykjavik University, Iceland. His research interests include CAD and modeling of microwave and antenna structures, simulation-driven design, surrogate-based optimization, space mapping, circuit theory, analog signal processing, evolutionary computation, and numerical analysis.